2009-12

# UAV digital tracking array design, development and testing

Pandya, Devieash James

Monterey, California: Naval Postgraduate School


# NAVAL POSTGRADUATE SCHOOL

**MONTEREY, CALIFORNIA**

# THESIS

**UAV DIGITAL TRACKING ARRAY DESIGN, DEVELOPMENT AND TESTING**

by

I-Hsiang Leslie Tan
Devieash James Pandya

December 2009

Thesis Advisor: David C. Jenn
Second Reader: Phillip E. Pace

**Approved for public release; distribution is unlimited**

THIS PAGE INTENTIONALLY LEFT BLANK

| REPORT DOCUMENTATION PAGE | | Form Approved OMB No. 0704-0188 |
|---|---|---|



| 1. AGENCY USE ONLY *(Leave blank)* | 2. REPORT DATE<br>December 2009 | 3. REPORT TYPE AND DATES COVERED<br>Master's Thesis |
|---|---|---|

| 4. TITLE AND SUBTITLE UAV Digital Tracking Array Design, Development and Testing | 5. FUNDING NUMBERS |
|---|---|
| 6. AUTHOR(S) I-Hsiang Leslie Tan and Devieash James Pandya | |
| 7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)<br>Naval Postgraduate School<br>Monterey, CA 93943-5000 | 8. PERFORMING ORGANIZATION REPORT NUMBER |
| 9. SPONSORING /MONITORING AGENCY NAME(S) AND ADDRESS(ES)<br>N/A | 10. SPONSORING/MONITORING AGENCY REPORT NUMBER |

**11. SUPPLEMENTARY NOTES** The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

| 12a. DISTRIBUTION / AVAILABILITY STATEMENT<br>Approved for public release; distribution is unlimited | 12b. DISTRIBUTION CODE |
|---|---|

**13. ABSTRACT (maximum 200 words)**

This thesis carried out the design and development of an integrated array and receiver processor that utilizes advanced techniques of Robust Symmetric Numeric System (RSNS) and monopulse Digital Beamforming (DBF) to accurately track a UAV using commercial-off-the-shelf (COTS) equipment. This was based on previous work done using virtual spacing RSNS and digital beamforming to extend the method to a functional six element array with direction finding and tracking capability. The six-element antenna array and direct-conversion receiver were developed and tested to retrieve a FM-modulated video signal encoded using the NTSC format. This thesis addresses system-level design tradeoffs, as well as hardware and software design, development and testing. A bench top test was conducted to test the functionality of the NTSC decoding and FM software modules developed and a comprehensive test was done in an anechoic chamber to characterize the array's capability in DF using RSNS and monopulse DBF.

| 14. SUBJECT TERMS<br>Phase Array, Digital Beamforming, RSNS, UAV Tracking, FM, NTSC, Direct Conversion, IQ | 15. NUMBER OF PAGES<br>190 |
|---|---|
| | 16. PRICE CODE |

| 17. SECURITY CLASSIFICATION OF REPORT | 18. SECURITY CLASSIFICATION OF THIS PAGE | 19. SECURITY CLASSIFICATION OF ABSTRACT | 20. LIMITATION OF ABSTRACT |
|---|---|---|---|
| Unclassified | Unclassified | Unclassified | UU |

NSN 7540-01-280-5500 Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std. 239-18

THIS PAGE INTENTIONALLY LEFT BLANK

**Approved for public release; distribution is unlimited**

**UAV DIGITAL TRACKING ARRAY**
**DESIGN, DEVELOPMENT AND TESTING**

I-Hsiang Leslie Tan
Civilian, ST Electronics, Singapore
B.Eng University of Surrey 2000

Devieash James Pandya
Major, Singapore Army
B. Eng National University of Singapore 2002

Submitted in partial fulfillment of the
requirements for the degree of

**MASTER OF SCIENCE IN ELECTRICAL ENGINEERING**

from the

**NAVAL POSTGRADUATE SCHOOL**
**December 2009**

Author: I-Hsiang Leslie Tan

Devieash James Pandya

Approved by: Professor David C. Jenn
Thesis Advisor

Professor Phillip E. Pace
Second Reader

Professor Jeffrey B. Knorr
Chairman, Department of Electrical and Computer Engineering

THIS PAGE INTENTIONALLY LEFT BLANK

# ABSTRACT

This thesis carried out the design and development of an integrated array and receiver processor that utilizes advanced techniques of Robust Symmetric Numeric System (RSNS) and monopulse Digital Beamforming (DBF) to accurately track a UAV using commercial-off-the-shelf (COTS) equipment. This was based on previous work done using virtual spacing RSNS and digital beamforming to extend the method to a functional six element array with direction finding and tracking capability. The six-element antenna array and direct-conversion receiver were developed and tested to retrieve a FM-modulated video signal encoded using the NTSC format. This thesis addresses system-level design tradeoffs, as well as hardware and software design, development and testing. A bench top test was conducted to test the functionality of the NTSC decoding and FM software modules developed and a comprehensive test was done in an anechoic chamber to characterize the array's capability in DF using RSNS and monopulse DBF.

THIS PAGE INTENTIONALLY LEFT BLANK

# TABLE OF CONTENTS

THIS PAGE INTENTIONALLY LEFT BLANK

# LIST OF FIGURES

# LIST OF TABLES

THIS PAGE INTENTIONALLY LEFT BLANK

# EXECUTIVE SUMMARY

In recent operations, there has been an increasing use of Unmanned Aerial Vehicles (UAVs) for real-time missions. UAVs used in these missions transmit the real-time data back to tactical field headquarters (HQs) for battlefield situation awareness and decision making. To continuously receive the signal from the UAV, the ground control station (GCS) will have to accurately track the aircraft using the antenna on the station. When the signal is received at the GCS, techniques are then applied to the signal to retrieve the imagery sent by the UAV.

There have been many technological advancements made in the field of monolithic integration, as integrated circuits (ICs) embrace more parts of RF systems and antennas, especially in the area of antenna arrays. In addition, and leveraging the advancement of digital technology, many processing tasks previously done via hardware are now done in the software domain.

The purpose of this research is to design and build an integrated array and receiver processor that utilizes advanced techniques of Robust Symmetric Numeric System (RSNS) and monopulse Digital Beam Forming (DBF) to accurately track a UAV using commercial-off-the-shelf (COTS) equipment for a low-cost solution. A six-element antenna array and direct-conversion receiver were developed to build a digital tracking array. The digital tracking array was also tested to retrieve a FM-modulated video signal encoded using the NTSC format. This thesis addresses system-level design tradeoffs, as well as hardware and software design, development and testing. The prototype array was tested in the NPS anechoic chamber to verify the tracking capability.

The focus on the hardware portion was to integrate all hardware modules developed for the complete system to work at optimum performance levels. Calculation for the system power budget was done to work out the required power levels. This required the careful calibration of the power levels provided across all modules, taking into account the optimum levels required by the inputs such as the demodulator LO drive power level. Care was also taken during the integration of the antenna array and receivers into a single chassis to ensure that high levels of workmanship is achieved so that no

failure should be attributed to workmanship problems such as a loose cable connection or insufficient power level during testing. This resulted in time savings as failures or inconsistencies in the test results can be quickly nailed down rather than spending additional effort to include the hardware during troubleshooting. Measurements were also done for the new two-element subarray antenna to check the measured performance against expected simulation results. Based on data obtained from antenna measurements and power budget calculations, calculation of the performance envelope for the system was presented.

The software for the antenna array was developed using a modular approach to facilitate testing and development, and to create avenues for future extension. Three separate modules were developed: NTSC Decoding, FM Demodulation and Monopulse Beamforming and Tracking. The NTSC Decoding module was successfully implemented for a single channel array, with the module able to decode incoming video signals and display the generated output video in monochrome on a terminal. The FM Demodulation Module was developed and tested successfully with sinusoidal input waveform for a single channel array. The Beamforming and Tracking Module used past research work done on RSNS and combined it with additional research conducted in monopulse DBF to form a module capable of using both these techniques to acquire and track a UAV using the six-element antenna array. A separate set of codes were implemented in a Matlab simulation to calculate expected theoretical values. These theoretical values were used to check against the performance of the software codes implemented in LabView.

The hardware and software solutions were integrated and tested using a bench top test setup where the receiver array was excited with analog phase shifters set to specific, calculated values to simulate the angle of arrival (AOA) of a signal. The receiver array then graduated to a more elaborate tracking test that was set up in the anechoic chamber to where different AOAs can be simulated through the rotation of a pedestal. Various scenarios such as different transmit powers were also tested. The antenna array was capable of acquiring and tracking the AOA of the incident waveform as the AOA was varied from $-30^{\circ}$ to $30^{\circ}$. The performance of the antenna array was tested at different levels of received power to investigate the performance capability of the system.

# LIST OF ACRONYMS AND ABBREVIATIONS

| | |
|---|---|
| ADC | Analog to Digital Converter |
| ADT | Automatic Detection and Track |
| AGC | Automatic Gain Control |
| AM | Amplitude Modulation |
| AOA | Angle of Arrival |
| AUT | Antenna Under Test |
| COTS | Commercial-of-the-shelf |
| DBF | Digital Beamforming |
| DC | Direct Current |
| DF | Direction Finding |
| EMI | Electromagnetic Interference |
| FOV | Field of View |
| FSK | Frequency Shift Keying |
| G | Gain |
| GCS | Ground Control Station |
| GPIB | General Purpose Interface Bus |
| HQ | Headquarter |
| HSYNC | Horizontal Sync Signal |
| FM | Frequency Modulation |
| I | In Phase |
| IC | Integrated Circuit |
| IF | Intermediate Frequency |
| IIP2 | Second Order Intercept Point |
| IRE | Institute of Radio Engineers |
| LNA | Low Noise Amplifier |
| LO | Local Oscillator |
| LPA | Low Power Amplifier |
| LPF | Low Pass Filter |
| MGTOW | Maximum Gross Take Off Weight |
| MTBF | Mean-Time Between Failures |

| | |
|---|---|
| NF | Noise Figure |
| NI | National Instruments |
| NPS | Naval Postgraduate School |
| NTSC | National television Standards Committee |
| Opo-Amp | Operational Amplifier |
| PCB | Printed Circuit Board |
| PCI | Peripheral Component Interconnect |
| PAL | Phase Alternating Line |
| PLL | Phased Lock Loop |
| PM | Phase Modulation |
| Q | Quadrature Phase |
| RF | Radio Frequency |
| RSNS | Robust Symmetric Numeric System |
| SEAD | Suppression of Enemy Air Defences |
| SECAM | Systeme Electronic Pour Avec Memoire |
| SL | System Loss |
| SNR | Signal to Noise Ratio |
| STT | Single Target Tracker |
| T | Temperature |
| T/R | Transmit/Receive |
| TWS | Track While Scan |
| UAV | Unmanned Aerial Vehicle |
| USAF | United States Air Force |
| VCO | Voltage Controlled Oscillator |
| VI | Virtual Instrument |
| VSYNC | Vertical Sync Signal |

# ACKNOWLEDGMENTS

We would like to thank our thesis advisor, Professor David Jenn, for your guidance, support, and patience during the development of our research and thesis, especially for your constant availability whenever we needed you. We would also like express our gratitude to Robert Broadston, Director of Microwave Laboratory. We are in your debt for all your advice and uncanny knack to solve problems. Appreciation also goes out to Professor Phillip Pace for advice and suggestions and Professor Frank Kragh for imparting to us your knowledge in signal modulation.

The co-authors, Leslie and James, would also like to thank ST Electronics (Satcom & Sensor Systems) and the Singapore Army respectively for their investment in our postgraduate education.

THIS PAGE INTENTIONALLY LEFT BLANK

# I. INTRODUCTION

## A. BACKGROUND

After years of being regarded as funny-looking model aircraft with limited military potential, Unmanned Aerial Vehicles (UAVs) are now the hottest commodity in the defence industry [1]. Currently, UAVs are being employed in an increasingly diverse range of military applications such as reconnaissance and attack missions. UAVs are viewed as force multipliers that can increase troop effectiveness. For example, one hovering UAV on surveillance duty could patrol the same area as ten or more human sentries due to its longer endurance. UAVs are seen as a key element in future combat systems. UAVs are accepted as a vital means to improve intelligence gathering, targeting and battlefield situation awareness while reducing the risks to military personnel, all at a lower cost. As an example, Northrop Grumman's RQ-4 Global Hawk (Figure 1), one of the largest UAVs with a maximum gross take-off weight (MGTOW) of up to twelve tons and a range of 13,500 nautical miles, costs around $14.8 million. The closest manned aircraft, in terms of characteristics, is the U-2, which would cost more than $40 million in 1999 dollars, and has less range and endurance [2].

Other than military applications, UAVs are also increasingly used in a number of civilian applications, such as firefighting, disaster monitoring (e.g., wildfires, earthquakes, tsunamis and cyclones), search and rescue, atmospheric observation and law enforcement.

Whether it is for a military or civilian application, UAVs are increasingly being preferred for the "3D"—dull, dirty and dangerous—missions. Some examples of "3D" missions are:

- Dull—maritime patrol work
- Dirty—monitoring contaminated or hazardous areas without risk to human life, such as in the Chernobyl disaster or monitoring wildfires
- Dangerous—high-risk missions like suppression of enemy air defences (SEAD)

Figure 1. USAF Global Hawk UAV (From [3])

At the tactical level, UAVs are being designed for real-time missions such as intelligence gathering, including operations related to urban warfare. Unlike the large Global Hawk, which flew in missions in the Iraq War and yet was controlled from the ground control station (GCS) in Florida, these tactical level UAVs are much smaller and have MGTOW of only a couple of hundred of kilograms (see, for example, Figure 2). Command and control of these UAVs are also made through portable tactical computers that are usually deployed in the field, some even requiring line-of-sight (LOS). UAVs used in these missions transmit the real-time data back to tactical field headquarters (HQs) for battlefield situation awareness and decision making.


Figure 2. Soldiers preparing a RQ-7B Shadow UAV for launch (From AFP/Getty Images)

In order for the UAV to carry out its mission, two other essential ingredients are required. The first is that a "pilot" is required, although some UAVs can be programmed to carry out their missions autonomously, and secondly, data links are required to transmit data back to the GCS. Hence, the GCS is also an indispensable part of the UAV system. As depicted in Figure 3, in order for the GCS to receive the signal from the UAV, the GCS is required to accurately track the aircraft using the antenna on the station. When the signal is received at the GCS, techniques are then applied to the signal to retrieve the imagery and other sensor data sent by the UAV. These help to reduce the complexity and time lag in the sensor-to-shooter link between the intelligence gathering and operational response.

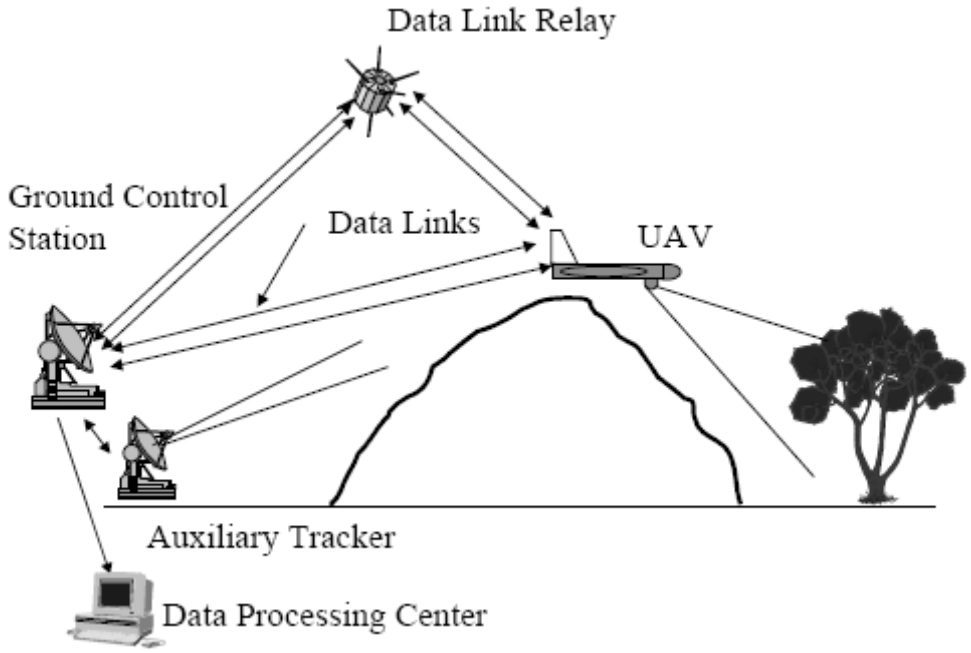



Figure 3. Elements of a UAV System (From [2])

There have been many technological advancements made in the field of monolithic integration, as integrated circuits (ICs) embrace more parts of RF systems, and antennas, especially in the area of antenna arrays. In addition, leveraging the advancement of digital technology, many processing tasks previously done via hardware are now done in the software domain. These have combined to enable GCSs to be built at relatively lower cost with phased array antennas, high performance integrated IC receivers and digital data processing capabilities. These GCSs are fast-deployable and portable because they are lightweight, robust and yet are still able to carry out their primary function of track, receive and data processing.

Phased array antennas are basically antennas that are scanned electronically via the adjustment of the weights for each individual antenna element. This results in fast and agile scanning. A digital array (Figure 4) employs a digital beamformer (computer processor) rather than a conventional microwave beamforming network. This allows multiple, simultaneous beams that can be produced to individually track multiple UAVs. Also, as these antennas are scanned electronically, there are no bulky and complicated

mechanical gimbals present to perform beam steering. As there is no mechanical wear and tear, the phased array antennas require almost no preventive maintenance. Power otherwise needed to power the mechanical gimbals is also eliminated.

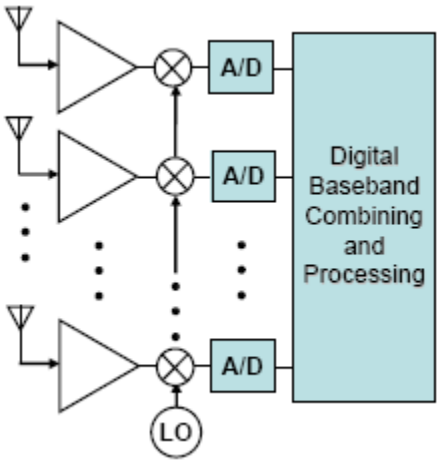



Figure 4. Multiple channels in a phased array receiver architecture

The monolithic integration of radio frequency (RF) and microwave circuits enabled miniaturization of circuits, which greatly reduced equipment size and power consumption (Figure 5). This also aided the development of phased array antennas as each element of the array can now be fed by individual receivers constructed using solid-state ICs. This vastly improves mean-time between failures (MTBF) in mission-essential functions as the failure of one receiver channel does cause failure to the entire system. Rather, "graceful degradation" only slightly impairs the performance of the system. For a single-channel receiver, failure of the receiver causes a catastrophic failure of the system as there is only a single point of failure. Monolithic integration also eliminated or reduced the performance inhibitors of the homodyne or direct-conversion receiver architecture, thus allowing its use.

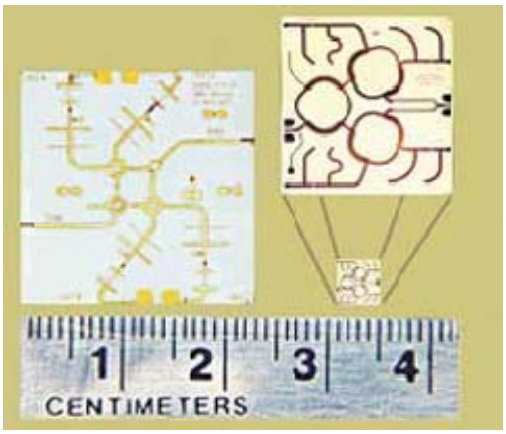



Figure 5. Although very different in size, these two devices can perform the same function in a microwave system (From [4])

In a digital phased array, after down-conversion, digital processing techniques can be applied to retrieve the original transmitted signal as well as to track the transmitting UAV. With technological advancements, these digital processing techniques can now deliver almost real-time information such as video to the GCS, as well as the processing of data to calculate the weights to be applied to each antenna element of the antenna array for beam steering. Ground troops could benefit from this since they no longer have to send out reconnaissance parties to get into close proximity in order to obtain intelligence. Instead, UAVs can be launched to get video imagery. These video images can also be broadcast to other units in the vicinity or back to the rear HQ for dissemination.

## B. PREVIOUS WORK

This research is part of an ongoing project that originally started with Bartee [5] in 2002 and continued with Burgstaller's [6] characterization of the critical commercial-of-the-shelf (COTS) hardware to reduce cost. The characterization of the COTS used in the project yielded valuable performance information. This information was used to build a demonstration transmit/receive (T/R) module for a distributed array radar [7].

Gezer [8] built on the information and implemented the processor in software using the LabView cRIO system to process downconverted data from the COTS hardware. Extensive simulation was also employed to fine-tune adjustments, ensuring

that the demonstrator built achieved the performance expected by the design simulations. Gezer also provided a good summary of the work done by previous students in his thesis.

The Robust Symmetrical Number System (RSNS) is an architecture that has been used for both analog and digital signal processing. Work was done by Lee [9] in developing a three channel phase interferometer using RSNS. He studied the concept, and built and tested the antenna array using relatively prime moduli for direction finding. Benveniste [10] extended this work and developed a single channel RSNS system with virtual spacing for direction finding. Using this technique, the RSNS moduli values used to implement the virtual spacing could be changed without changing the actual element spacing.

## C. SCOPE OF RESEARCH

The purpose of this research is to design, build and test an antenna array and digital receiver that utilizes advanced techniques of Robust Symmetric Numeric System (RSNS) and Digital Beamforming (DBF) to accurately acquire and track a UAV signal using the monopulse technique. Hardware is implemented using COTS equipment for a low-cost solution.

Optimum power levels for the direct-conversion receiver were determined and empirically verified to ensure optimum performance. The antenna array and receivers were then integrated into a single chassis and tested again to verify performance. The six-element antenna array was tested in an anechoic chamber to ensure that the expected performance was obtained.

Software modules were developed and tested to retrieve a FM-modulated video signal encoded using the NTSC format. The tracking algorithm was also implemented in LabView using the monopulse technique to enable accurate tracking of the UAV. A separate set of codes were implemented in a Matlab simulation to calculate expected theoretical values. These theoretical values were used to check against the performance of the software codes implemented in LabView.

As an initial test, the receiver array was excited with analog phase shifters set to specific, calculated values to simulate the angle of arrival (AOA) of a signal. The receiver array then graduated to a more elaborate test that was set up in the anechoic chamber where different AOAs can be simulated through the rotation of a pedestal. The tracking test was also done on this set up as described in Chapter VI. Various scenarios such as different transmit powers were also tested.

## D. ORGANIZATION OF THESIS

Chapter II discusses the reception of Frequency Modulation (FM) signals from the UAV, the types of video signals and the NTSC format used to encode the video signals. In addition, this chapter also discusses the modulation technique used in FM, the mathematical derivation and the building blocks of the FM demodulator.

Chapter III presents the tracking principles and different modes for a tracking antenna. Different tracking techniques such as sequential lobing, conical scan and monopulse tracking are also presented. The Robust Symmetric Numeric System (RSNS) theory and its application to direction finding is also briefly covered in this chapter. Finally, this chapter explains the key concepts behind Digital Beam Forming (DBF) and Phase Interferometry.

Chapter IV covers the benefits of direct-conversion receiver architecture over that of the ubiquitous superheterodyne architecture, the mathematical derivation for direct-conversion of a radio frequency (RF) signal to baseband and the problems associated with direct-conversion and methods to minimize them.

Chapter V overviews the entire receiver system, highlighting the major building blocks in both the hardware and software portions of the receiver. Functional descriptions of these major building blocks are also provided in detail. Section C in this chapter presents the preliminary performance analysis of the system, where some of the key performance indicators such as the maximum operating range are estimated.

Chapter VI presents the verification and validation of each of the hardware and software components developed for the receiver system. Each of the test results of the

individual components such as the antenna and software modules are presented. This chapter also describes the integration of the hardware and software components for the purpose of acquisition and tracking, details the bench top test and showcases the test results from the anechoic chamber.

Chapter VII summarizes and analyzes the results obtained and provides recommendations for the way forward.

The Appendix contains the Matlab codes written for the various calculations and plots presented in the analysis.

THIS PAGE INTENTIONALLY LEFT BLANK

# II. RECEPTION OF THE UAV FM VIDEO SIGNAL

When developing a UAV tracking array, there are two primary design aspects that are considered: tracking and acquisition of the UAV signal, and reception of data transmitted from the UAV. In this chapter, the main focus is on the reception of data from the UAV. For the current application (a typical reconnaissance UAV), this data is in the form of a signal that contains the video data captured by the onboard camera. This continuous data is encoded using a particular analog video format and modulated on a carrier frequency prior to transmission. At the receiver, the received signal is then unwrapped using the corresponding demodulation scheme, as was used on the UAV. Finally the signal is decoded to retrieve the captured video data. In this chapter, the theory of demodulation and video decoding procedures are addressed in detail, with emphasis placed on the techniques used in the design of the proposed antenna, which is detailed in Chapter V.

## A. VIDEO SIGNALS

Video signals are widely used today in the transmission of video data in many different media (television, internet, etc.). Video signals can be broadly classified into two main categories: analog and digital. Analog video signals are video signals that are represented by a fluctuating voltage signal in the form of an analog waveform. These signals form a continuous waveform that encodes video information into the resulting waveform. Digital video signals refer to a category of signals that are represented by computer readable binary numbers that describe a finite set of colors and luminance levels [11]. Historically, most broadcast transmission of video signals in free space is based on the analog format. Analog video signals are still commonly used for the transmission of video data from the UAVs to the GCS (for example, the ScanEagle UAV system [12]). For the purpose of this thesis, analog video signal will be studied in detail for its implementation in the receiver.

### 1. Analog Video Signals

There are several analog video formats in use today, namely, National Television Standards Committee (NTSC), Phase Alternating Line (PAL) and Systeme Electronic Pour Avec Memoire (SECAM). These formats have their roots in the different standards that have been developed for analog television broadcast in the different regions of the world.

The main difference between the formats stems from the power frequencies used in the world. There are two main frequencies used: 50 Hz and 60 Hz. Early designers of TVs used these main frequencies as a source for the *field timing reference signal* (i.e., the indicator for the TV to receive the next picture in a stream of images). Due to this difference, there are two main TV systems, one based on 25 frames per second (50 Hz) and the other based on 30 frames per second (60 Hz) [13]. Most countries with 60 Hz power supply use the NTSC format that was developed in the United States of America. PAL and SECAM were developed for countries with a 50 Hz power supply, although the former is used in several countries with 60 Hz as well. See Table 1 for details on each of the formats [14].

Table 1. Analog video standards (From [14])

| Format | Used | Mode | Signal Name | Frame Rate (frame/sec) | Vertical Line Resolution | Line Rate (lines/sec) | Image Size |
|---|---|---|---|---|---|---|---|
| NTSC | U.S., Japan | Mono | RS-170 | 30 | 525 | 15,750 | 640x480 |
| | | Color | NTSC Color | 29.97 | 525 | 15,734 | |
| PAL | Europe (except France), Asia | Mono | CCIR | 25 | 405 | 10,125 | 768x576 |
| | | Color | PAL Color | 25 | 625 | 15,625 | |
| SECAM | France, Europe | Mono | | 25 | 819 | 20,475 | N.A |
| | | Color | | 25 | 625 | 15,625 | |

## 2. Fundamentals of a Video Signal

For an analog signal to be displayed on a screen such as a monitor or TV, there needs to be a system for the signal to be converted from signal value to pixel data for display. For video scanning, a system of left to right, top to bottom is used. Figure 6 shows this scanning process.

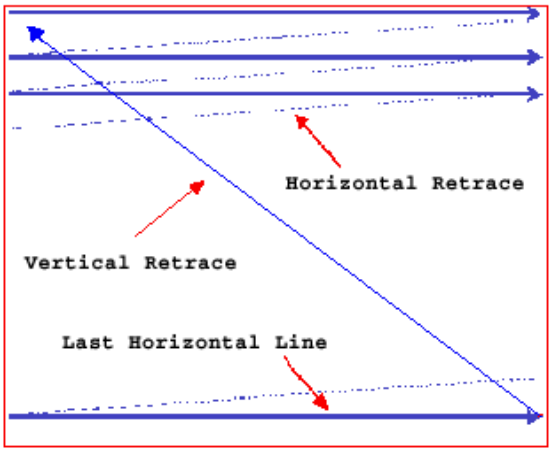



Figure 6. Video scanning system (From [14])

Most systems today use interlaced scanning (Figure 7), where an image is divided into two fields, one containing even-numbered horizontal lines and the other containing odd-numbered lines. The odd field is always scanned before the even field and this process is repeated for every frame. For the NTSC format, the image refreshes at 30 frames per second, which is perceivable by the human eye. With interlaced scanning, the image is updated at 60 fields per second and this ensures the flicker is not perceived by the human eye during updates.

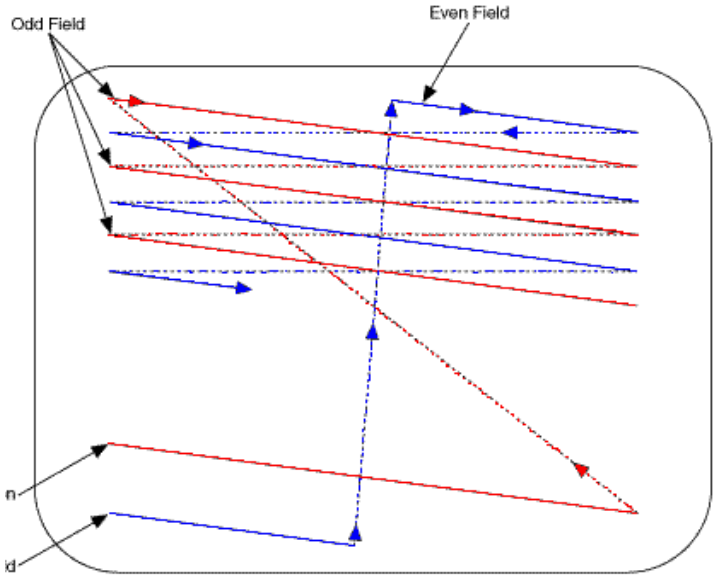



Figure 7. Interlaced scanning (From [15])

An analog video signal contains information on the intensity and timing of the video which, when used together, allows the re-creation of the video. There are three main signal components that are embedded within a single video signal: luma (luminance), chroma (chrominance) and the synchronization signal.

The luma signal contains the information on the intensity of the video image. It provides a reference for the brightness of the image. It represents an image without the necessary color information.

The chroma signal contains the color information of the image. This information is separate from that carried by the luma signal. In addition to the chroma signal, there is

a color burst located in the video signal that provides the phase and amplitude reference for the color. When these are used in conjunction with the luma signal, a complete colored image is produced.

The synchronization (sync) signal controls the scan lines on a display screen (e.g., monitor). This sync signal consists of several sync signals such as the Horizontal Sync signal and the Vertical Sync. Horizontal Sync signal dictates when the image should be scanned from the left side while Vertical Sync signal dictates when the image should be scanned from the top of the image [16]. These sync signals are covered in detail in a later section.

A monochrome composite signal (also known as a Y signal) consists of the luma signal and the synchronization signal. The chroma signal is called a C signal. A color composite signal consists of both Y and C signals. See Figure 8 and Figure 9, respectively, for an example of a monochrome composite signal and a color composite signal. The unit used to define video levels within a signal is the IRE (Institute of Radio Engineers). The blanking level is used as 0 IRE which usually corresponds to 0 V as well. White level is set to 100 IRE, while black differs for the different video standards. NTSC represents black at 7.5 IRE while NTSC and PAL align black at 0 IRE.

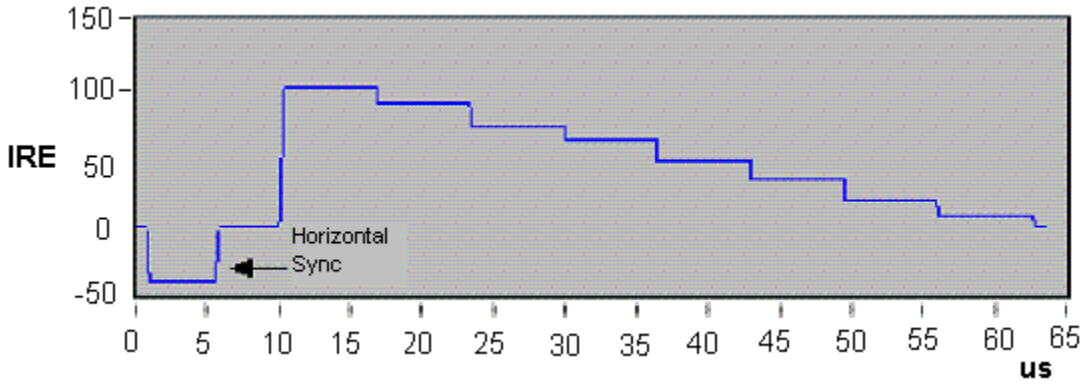



Figure 8. Y signal with Horizontal Sync Signal and varying luma levels (From [17])

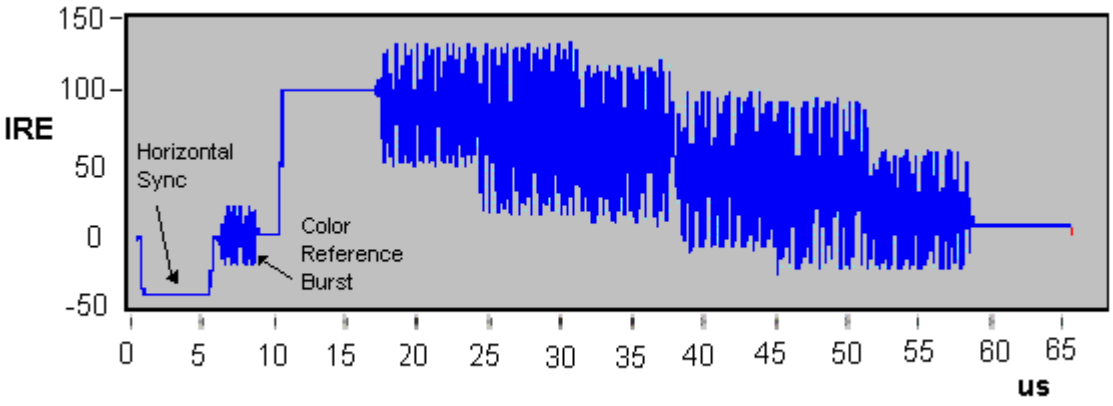



Figure 9. Composite color signal (From [17])

### 3. Horizontal Video

Each horizontal video line consists of the following components: horizontal sync signal, back porch, active pixel region and front porch. This is shown in Figure 10.

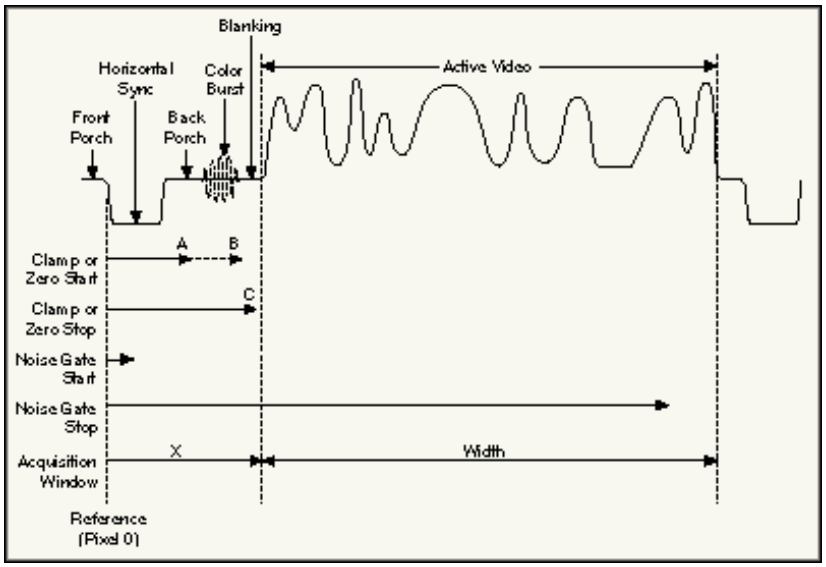



Figure 10. Horizontal video line (From [15])

The horizontal sync signal (HSYNC) denotes the beginning of a new line. During scanning, HSYNC indicates to the display that the next stream of pixels is to be

represented on the next line. For monochrome signals, dc components in the video signal are removed during the clamping interval that occurs on the back porch. For composite color signals, clamping interval occurs during the HSYNC pulse as the back porch is used for the color burst. The active pixel region contains the data for the video image.

In addition to HSYNC pulses, a video signal also contains vertical sync (VSYNC) pulses to perform a vertical retrace and scan the next field in a frame. Figure 11 shows a simple example of a frame with six lines.

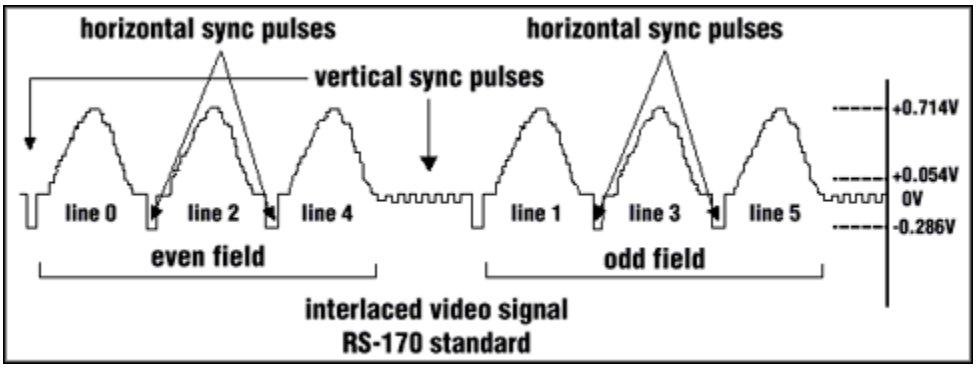



Figure 11. Interlaced scanning of a six line video signal[1] (From [14])

In Figure 11, the even field is scanned first. At the end of each line, a HSYNC pulse is used to indicate the start of a new line. At the end of the even field, several VSYNC pulses are inserted to indicate the start of a new field and the process is then repeated for the odd field.

## B. NTSC FORMAT

For videos based on the NTSC standard, each video frame is made up of two color fields or 525 lines. The first 20 lines consist of VSYNC pulses within the field blanking period. The VSYNC pulses here indicate the beginning of a new frame [18]. Color field two begins in the middle of line 263 (20 lines for VSYNC pulses, 262 lines of

[1] Figure 11 shows the interlaced scanning of a six line video signal. In the actual format, the odd field will be scanned first before the even field. The figure is only meant to show the concept of interlaced scanning.

video information) with another 20 lines of VSYNC pulses to indicate the next field in the frame with the remaining 262 lines containing video data. The format of the video data is illustrated in Figure 12.

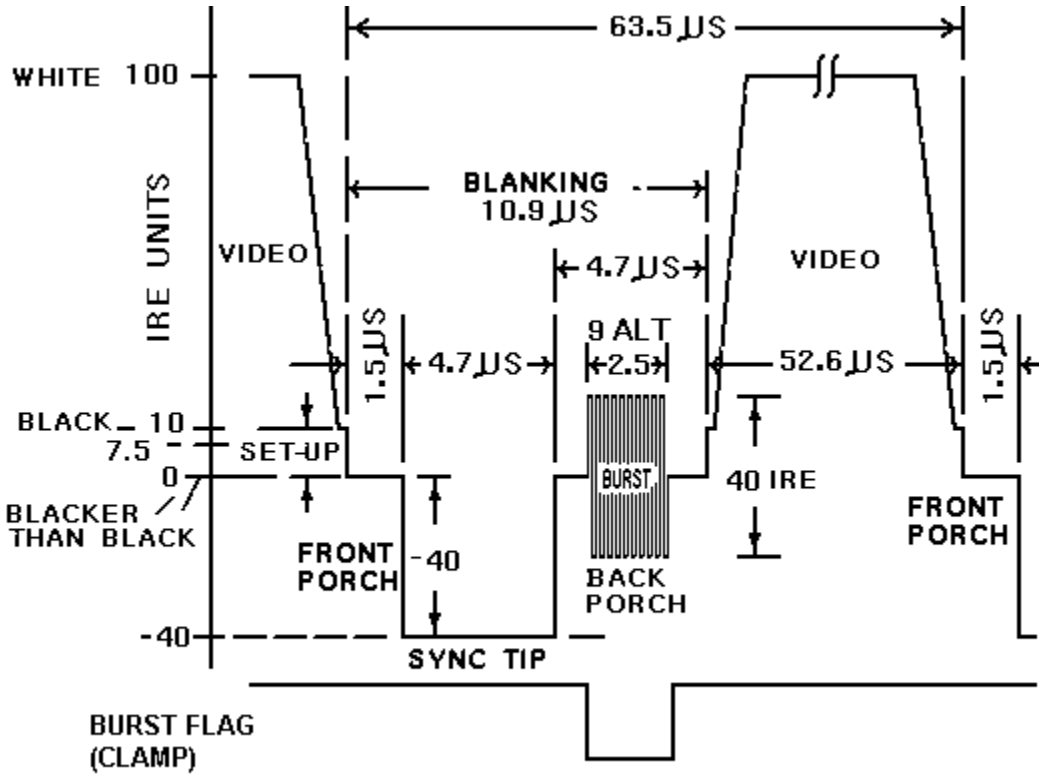



Figure 12. Detailed timing of a video line (From [11])

The horizontal size of the image obtained from an analog image capture device is dependent on the rate of frames sampled of the horizontal video line. This rate is determined by the vertical line rate and the specifications of the device. To avoid distortion, the video line must be sampled at a rate that ensures the horizontal active video region is divided into the correct number of pixels.

Some important parameters for NTSC are:

lines/frame = 525 (485 for data, 40 for VSYNC pulses)
NTSC vertical line rate = 15,750 Hz
Active video duration = $52.66\mu$sec
Number of active pixels/line = 640
Pixel rate = 640 pixels/line / $52.66\mu$sec/ line
$=12.15\times10^{6}$ pixels/sec

These values are used in Chapter VI in the development of the NTSC decoding software.

## C. SIGNAL MODULATION

When transmitting message signals over long distances through an unbounded medium such as air, message signals are often modulated on a carrier frequency. Generally modulation refers to the process by which low-frequency signals are impressed upon high-frequency carrier signals. There are several reasons why signals are modulated prior to transmission. This includes ensuring multiple access in a medium through bandwidth segregation, ensuring frequency of transmission facilitates physical size of antennas for transmit/receive, and selecting a transmitting frequency other than the message frequency to facilitate propagation in a given medium.

In an analog signal, there are three properties (amplitude, frequency and phase) that can be varied and used to modulate a message signal. Amplitude, frequency and phase modulation (AM, FM and PM respectively) are the three main types of analog modulation schemes. Frequency and phase modulation are both considered angle modulation. Angle modulation was first introduced in 1931 as an alternative to amplitude modulation. Today, angle modulation is used in commercial broadcasting, microwave and satellite communication systems.

When a UAV is on a reconnaissance mission, it obtains video footage via its onboard image capturing device before transmitting this data to a base station on the ground. This video signal is FM modulated on a carrier frequency and then transmitted.

At the base station, the received signal is down converted to base band and demodulated to retrieve the original signal. For the purpose of this thesis, emphasis is placed on the FM and its associated demodulation procedure.

### 1. Frequency Modulation (FM)

Angle modulation is the process where the angle of a waveform is varied with time. The waveform is expressed mathematically as

$$s(t) = V_c \cos[\omega_c t + \theta(t)] \tag{2.1}$$

where $s(t)$ = angle modulated waveform

$V_c$ = peak carrier amplitude

$\omega_c$ = carrier radian frequency

$\theta(t)$ = instantaneous phase deviation (rad)

For angle modulation, it is necessary for $\theta(t)$ to be a function of the modulating signal. If $v_m(t)$ is the modulating signal, angle modulation is given by the following [19]:

$$\theta(t) = F[v_m(t)] \tag{2.2}$$

where $v_m(t) = V_m \sin(\omega_m t)$

$F[\cdot]$ denotes a functional dependence on the argument

$\omega_m$ = radian frequency of the modulating signal ( $2\pi f_m$ rad/s )

$f_m$ = modulating signal frequency

$V_m$ = peak amplitude of the modulating signal

For FM, the frequency of the carrier is directly varied by the modulating signal. As a result of the variation of the carrier frequency, the phase of the waveform is also varied. This means that when FM is performed, PM occurs as well.

FM is defined [19] as the varying of frequency of the waveform proportional to the amplitude of the modulating signal at a rate equal to the frequency of the modulating

signal. Figure 13 shows an example of a FM signal in the frequency domain. The frequency is shifted in magnitude and direction corresponding to the amplitude and polarity of the modulating signal ($V_m$) and the rate of change of the frequency is equal to the frequency of the modulating signal. The displacement of the carrier frequency from its original frequency is known as the frequency deviation ($\Delta f$). The instantaneous frequency deviation, $\theta'(t)$ is the change of the carrier frequency of the signal and is defined as the first derivative of the instantaneous phase deviation, $\theta(t)$,

$$\Delta f = \theta'(t) \ {}^{\text{rad}}\!/_{\text{s}} \tag{2.3}$$

For FM, the relation of $\theta(t)$ to $v_m(t)$ is given by

$$\theta(t)_{FM} = F[v_m(t)] = D\int_{-\infty}^{t} m(\alpha)\,d\alpha \tag{2.4}$$

where $D$ is a constant known as the deviation sensitivity measured in radians/volt-sec or Hz/volt, $\int_{-\infty}^{t} m(\alpha) = v_m(t)$.

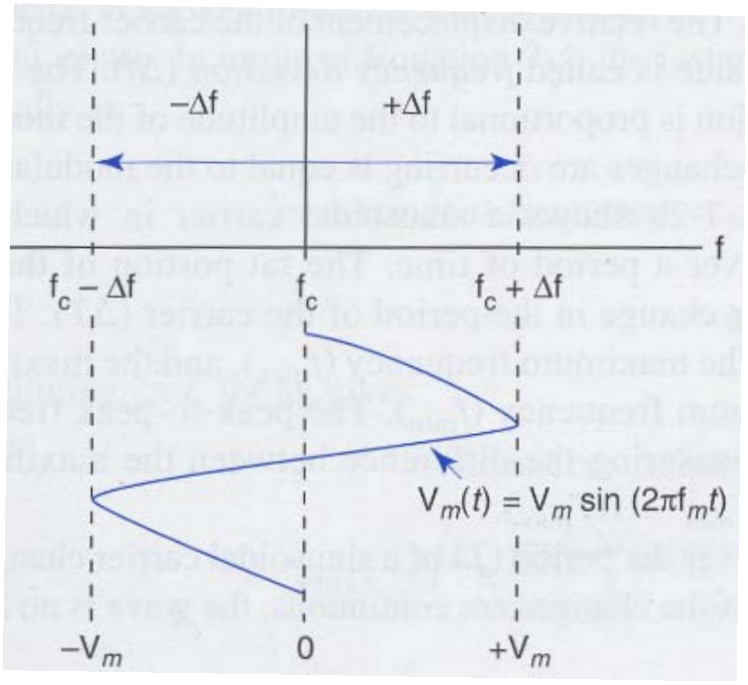



Figure 13. An example of a FM signal in the frequency domain (From [19])

Taking a derivative of both sides of Equation (2.4) with respect to time leads to the following equation:

$$\frac{d(\theta(t)_{FM})}{d(t)} = \frac{d(D\int_{-\infty}^{t} m(\alpha)\,d\alpha)}{d(t)} = Dm(t) \tag{2.5}$$

## 2. FM Demodulation Techniques

When a signal with FM is received, it then will be placed through a demodulation process to retrieve the original message signal. Of interest in this research is the signal received from a UAV with video information encoded using the NTSC format. From [9], the parameters of the FM video are based on a bandwidth of approximately 6 MHz for video and audio transmission. FM demodulation is treated mathematically in this section.

The received signal, $s(t)$, is represented in terms of in-phase (*I)* and quadrature (*Q)* components. This is done because the proposed array architecture uses *I* and *Q* for the single-stage down conversion from RF to baseband. Using the in-phase and quadrature representation:

$$\begin{aligned} s(t) &= V_c \cos\left(2\pi f_c t + D\int_{-\infty}^{t} m(\alpha)\,d\alpha\right) \\ &= V_c \cos\left(2\pi f_c t + 2\pi\Delta f t + D\int_{-\infty}^{t} m(\alpha)\,d\alpha\right) \\ &= V_c \cos\left(2\pi\Delta f t + D\int_{-\infty}^{t} m(\alpha)\,d\alpha\right)\cos(2\pi f_c t) \\ &\quad - V_c \cos\left(2\pi\Delta f t + D\int_{-\infty}^{t} m(\alpha)\,d\alpha\right)\sin(2\pi f_c t) \end{aligned} \tag{2.6}$$

where

$$I(t) = \cos\left[\left(2\pi\Delta ft\right) + D\int_{-\infty}^{t} m(\alpha)\,d\alpha)\right]$$
$$Q(t) = \sin\left[\left(2\pi\Delta ft\right) + D\int_{-\infty}^{t} m(\alpha)\,d\alpha)\right] \qquad (2.7)$$

*I(t)* is the in-phase component and *Q(t)* is the quadrature component.

In frequency modulation, the information of the message signal is encoded in the change of frequency of the modulated signal. As discussed earlier, a change in frequency leads to a change in phase of the signal. To retrieve the original signal, the rate of change of the frequency is obtained by computing the first derivative of the *I(t)* and *Q(t)*:

$$\frac{dI(t)}{dt} = -\sin\left[\left(2\pi\Delta ft\right) + D\int_{-\infty}^{t} m(\alpha)\,d\alpha)\right]\left(2\pi\Delta ft + Dm(t)\right)$$
$$\frac{dQ(t)}{dt} = \cos\left[\left(2\pi\Delta ft\right) + D\int_{-\infty}^{t} m(\alpha)\,d\alpha)\right]\left(2\pi\Delta ft + Dm(t)\right) \qquad (2.8)$$

Finally to retrieve the message signal, the expression $I\frac{dQ(t)}{dt} + Q\frac{dI(t)}{d(t)}$ is evaluated:

$$\begin{aligned} & I\frac{dQ(t)}{dt} + Q\frac{dI(t)}{d(t)} \\ &= \left(2\pi\Delta ft + Dm(t)\right)\cos^2\left[\left(2\pi\Delta ft\right) + D\int_{-\infty}^{t} m(\alpha)\,d\alpha)\right] \\ &\quad + \left(2\pi\Delta ft + Dm(t)\right)\sin^2\left[\left(2\pi\Delta ft\right) + D\int_{-\infty}^{t} m(\alpha)\,d\alpha)\right] \\ &= \left(2\pi\Delta ft + Dm(t)\right) \\ &\quad \overbrace{\left[\cos^2\left[\left(2\pi\Delta ft\right) + D\int_{-\infty}^{t} m(\alpha)\,d\alpha)\right] + \sin^2\left[\left(2\pi\Delta ft\right) + D\int_{-\infty}^{t} m(\alpha)\,d\alpha)\right]\right]}^{1} \\ &= \left(2\pi\Delta ft + Dm(t)\right) \end{aligned} \qquad (2.9)$$

The expression $I\frac{dQ(t)}{dt} + Q\frac{dI(t)}{d(t)}$ is a simple manner through which the original message is retrieved from the *I* and *Q* representation of the modulated signal. The retrieved component, $2\pi\Delta ft$, is the potential dc offset due to a difference between the

carrier frequency and the local oscillator frequency used during the down conversion of the signal to baseband (or intermediate frequency (IF)). This is in essence how a simple FM demodulator can be configured. In practical implementations of FM demodulators, three methods to demodulate FM signals were studied and discussed in [20].

### *a. Digital Phase Lock Loop (PLL) Quadrature Detector*

A PLL consists of three main components: phase comparator, low pass filter (LPF) and voltage controlled oscillator (VCO). In circuits with a PLL, when an input signal is applied to the circuit, the phase comparator generates an error voltage signal proportional to the difference between the VCO and the input signal. This error voltage is then applied to the input of the VCO. When done recursively, this reduces the difference between the input signal and VCO.

### *b. Local Oscillator with No Feedback*

A system with a local oscillator with no feedback is similar to a PLL with the exception that there is no error voltage generated or feedback to the VCO. In this approach, it is assumed that the oscillator has the same frequency as the input signal. If both the input frequency and input signal are equal, the message can be demodulated accurately.

### *c. Bilotti's Quadrature Demodulator*

A modulated signal's deviation from the carrier frequency represents the FM modulation of the signal. To retrieve the original signal, the frequency deviation of the signal is converted to either phase or amplitude and the change in phase or amplitude is detected. In the quadrature demodulator (also called Bilotti's method), the modulated carrier is passed through a circuit that shifts the signal by $90^0$ at the center frequency. The direction of frequency deviation will cause the phase shift to be either greater or smaller than $90^0$. The circuit simultaneously converts the FM to amplitude modulation (AM), so an amplitude detector could also be used. Bilotti's demodulator uses a phase shift network and a phase detector. The output of the detector is applied to either an AM or PM detector.

In [20], simulations were run to determine the effect of demodulation using the three means of detection stated above. Simulations were run for digital beamforming with the introduction of phase errors between array channels to simulate practical scenarios operating the antenna. It was concluded that the FM detector, LO with no feedback, performed as well as the other two detectors with less circuitry used. This detector was thus selected as a choice of FM detection in the implementation of FM demodulation in this thesis.

In this chapter, the fundamentals of NTSC decoding and FM demodulation were covered. In the next chapter, general tracking arrays and techniques will be discussed in detail.

THIS PAGE INTENTIONALLY LEFT BLANK

# III. TRACKING ARRAYS AND TECHNIQUES

In this chapter, there are two main concepts covered in detail. In Section A, the main principles of a tracking array antenna are covered. In Section B, a brief overview of Robust Symmetric Number System (RSNS) direction finding is covered. Finally, in Section C the key concepts of Digital Beamforming and Phase Interferometry are discussed. These sections form the basis of the antenna system proposed in this thesis.

## A. TRACKING ANTENNA

### 1. Tracking Principles

When tracking for radar, the target's angle and range are determined. For this particular application only angle tracking is of interest. There are many types of tracking principles that can be broadly classified into the four categories described in the following sections. They are discussed in terms of radar, but the comments also apply to the tracking of a signal from a UAV. Equation Chapter (Next) Section 1

#### *a. Single-Target Tracker (STT)*

This category of tracker describes the continuous tracking of a target at relatively high data rate. The antenna follows the target by employing a closed loop servo system to obtain the angle-error signal between the target location and antenna beam direction. Radars that utilize STT are usually designed for tracking of aircraft or used together with military weapon control system.

#### *b. Automatic Detection and Track (ADT)*

Unlike the STT, the ADT has a lower rate of target observations. The rate of these observations is dependent on the time taken for the antenna to make a rotation. The tracking of the target is done open loop and thus the antenna position is not determined by the data track.

### *c. Track While Scan (TWS)*

TWS is the category of tracking that maintains track of a number of targets within a sector of coverage. This is very similar to ADT and is considered today as a subset of the ADT category.

### *d. Phased Array Radar Tracking*

Phased array radars are the latest development of radars and represent the ability to track a large number of targets with a high data rate through the use of electronically steered phased array. With electronically steered beams, the direction of scan can be rapidly switched to the different target locations and thus, with adequate time sharing between targets, the radar is able to maintain a relatively high data rate for each target. Much of the work in this thesis will be focused on phased array tracking.

## 2. Tracking Techniques

In [8], it was noted that there are three main types of tracking techniques commonly employed in radars. These techniques are: (1) sequential lobing, (2) conical scan and (3) monopulse. These techniques are included here to give a holistic understanding of the subject matter. Additional emphasis is given to monopulse tracking as the concept will be used in the following section on digital beamforming.

### *a. Sequential Lobing*

In sequential lobing, a single beam is alternated between two separate beam directions to obtain an angle measurement. Figure 14 illustrates this concept. The beam is switched between position 1 and position 2 in the attempt to align the target on the switching axis. If the target is not on that axis, the difference in voltage values obtained from the two beam positions will give displacement of target from the switching axis.

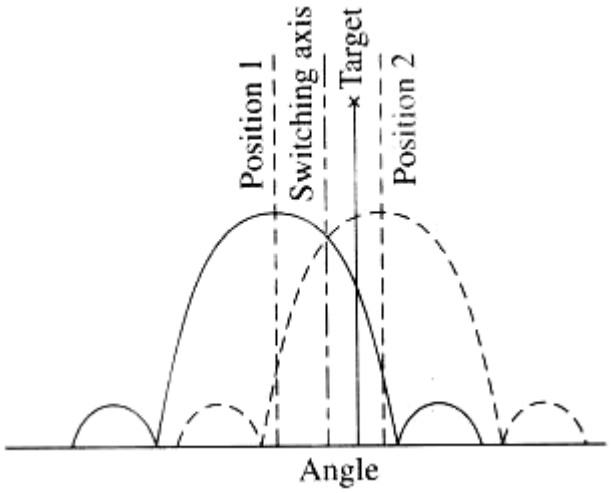



Figure 14. Lobe switching antenna patterns (From [21])

### *b. Conical Scan*

In conical scan, the antenna beam is rotated on an axis of rotation. The rotation of the beam with the target aligned on the axis of rotation results in the radar receiving equal amplitude echo responses as the beam is rotated. If there is an unequal echo, the location of the target can be determined relative to the axis of rotation. Figure 15 illustrates conical scan tracking. With the target located at A, the conical scan will determine the phase difference between the axes between A and the rotation axis of the beam. This difference is then applied to the rotation axis.

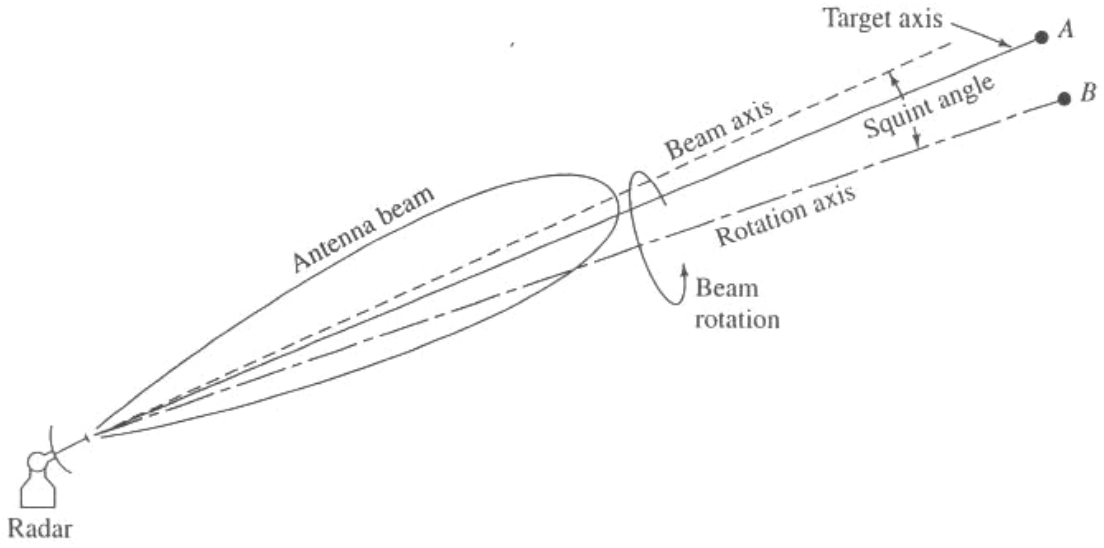



Figure 15. Conical scan tracking (From [22])

### *c. Monopulse Tracking*

The tracking techniques mentioned above utilize multiple signal samples to determine the error signal to effectively steer the antenna beam to the direction of the target. For monopulse tracking, only a single sample is used to track the target. It obtains the location of a target by comparing signals received in two or more simultaneous beams [21]. This form of tracking is considered to have a higher accuracy compared to time-shared tracking systems such as sequential lobing and conical scan.

There are two main methods by which the angular measurement can be made: amplitude-comparison monopulse and phase comparison monopulse.

For amplitude comparison monopulse two beams, pointing in slightly different directions, are created simultaneously. This system utilizes the sum and difference of these patterns to track the target. On transmit, only the sum of the patterns are used while on receive, both the sum and difference patterns are used. Similar to sequential lobing, the main idea is to position the null of the difference pattern in the direction of the target. The angular error is obtained by the offset from the null on the

difference pattern. Figure 16 shows a section of the difference pattern (normalized by the peak of the sum pattern). If the target is not at boresight, the result of the difference pattern at that scan angle will be non zero. The slope in the linear region is used to create the error signal. This error signal is given by the equation:

$$\frac{\Delta}{\Sigma} = K\theta \tag{3.1}$$

where $K$ is the monopulse slope constant. The monopulse slope constant can be determined from measurement or simulation.

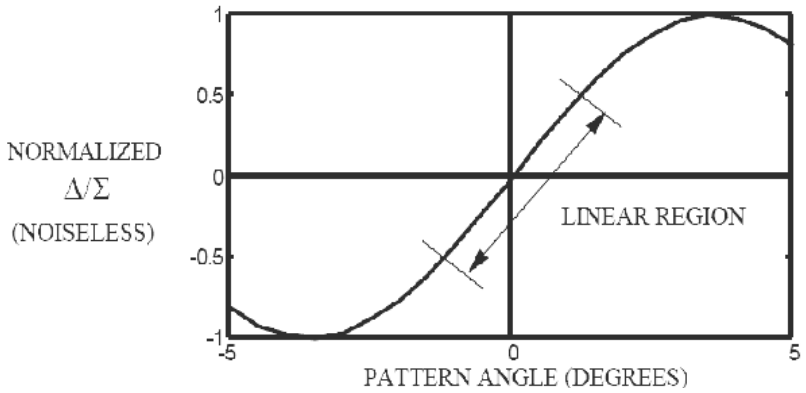



Figure 16. Difference pattern where scan angle is at boresight (From [23])

For phase comparison monopulse, similar to amplitude comparison monopulse, two beams are used to obtain the angle measurement. However in this case, both beams are pointing in the same direction and create an interferometer pattern. The error signal in this method is generated through the phase difference of the signal received by each beam.

## B. ROBUST SYMMETRIC NUMERIC SYSTEM (RSNS)

The RSNS has been applied to direction finding (DF). The primary function of RSNS DF for this application is acquisition of a signal. Once acquired, monopulse tracking can commence.

### 1. RSNS Theory

The main concept behind the use of RSNS in direction finding is the mapping of a symmetric waveform at an antenna into a RSNS sequence. A RSNS sequence is represented as follows:

$$x_m = [0,1,2,...,m-1,m,m-1,...,2,1] \tag{3.2}$$

where $x_m$ is a row vector and *m* is the modulus used and is a positive integer.

In an *N*-channel RSNS sequence, each element in $x_m$ is repeated *N* times. The basic form of the RSNS is given in the equation:

$$\{x\} = \begin{cases} \left\lfloor \dfrac{h - s_i}{N} \right\rfloor, & s_i \le h \le Nm_i + s_i + 1 \\ \left\lfloor \dfrac{2Nm_i + N - h + s_i - 1}{N} \right\rfloor, & Nm_i + s_i + 2 \le h \le 2Nm_i + s_i - 1 \end{cases} \tag{3.3}$$

where *h* is an integer with a value $0 \le h < m$ and $s_i$ is the sequence shift value.

With a set of selected moduli, a sequence for an *N* channel RSNS system can be developed using the equations above. The dynamic range of the sequence is defined as the longest sequence of unique vectors formed in the sequence. In direction finding, dynamic range is an important factor as this range will be mapped into the field of view (FOV) of the antenna and will thus contribute to the angular accuracy of the sequence in tracking within that FOV. For example, if the dynamic range of a sequence is 30 and the FOV is $180^\circ$, the angular resolution using the RSNS sequence will be $180/30 = 6^\circ$ at broadside.

Dynamic range can be computed for a two-channel RSNS sequence, with moduli $m_1$ and $m_2$, using the following equations:

For $m_1 > 3$ and $m_2 = m_1 + 1$,

$$\hat{m} = 3(m_1 + m_2) - 6 \tag{3.4}$$

For $m_1 > 5$ and $m_2 = m_1 + 2$,

$$\hat{m} = 3(m_1 + m_2) - 7 \tag{3.5}$$

For $m_1 > 5$ and $m_2 = m_1 + C$, where $C > 3$,

$$\hat{m} = 4m_1 + 2m_2 - 2 \tag{3.6}$$

As can be seen, the above cases define the computation of dynamic range for a two-channel RSNS sequence. In [24], a three-channel RSNS sequence was proposed and analyzed.

Once the dynamic range of the RSNS sequence has been determined, the number of folds within each modulus is found. This is given by the equation:

$$n = \frac{\hat{m}}{2Nm_i}, i = 1, 2..., N \tag{3.7}$$

Next, the required spacing between each channel is computed to achieve the number of folds found using Equation (3.7). This is computed using the equation:

$$d_i = n_i \frac{\lambda}{2} \tag{3.8}$$

To develop the different bins for the RSNS, threshold voltages have to be computed for each of the individual bins. Since each bin is comprised of a value from each modulus selected, $m_i - 1$ threshold values will be computed for each channel. These thresholds are computed using the following equation:

$$V_{j,m_i} = \cos\left(\frac{m_i - j + \frac{1}{2}}{m_i}\pi\right), 1 \le j \le m_i \tag{3.9}$$

The final step in developing a RSNS system is the computation of a phase adjustment term, $\varphi_i$ for each channel. Each modulus will not be symmetric about the center of the dynamic range and thus $\varphi_i$ is used to shift the center of the dynamic range

to the broadside angle of arrival [24]. These values align the folding waves at each channel. With the addition of the phase adjustments, the output voltage at each channel is given by:

$$V_i(\theta) = \cos(kd\sin(\theta) + \varphi_i) \tag{3.10}$$

### 2. RSNS Virtual Spacing

The steps in Section 2 detail the development of a RSNS system for direction finding. Practically, the channel spacing computed using the equations may not be achievable due to limitations such as the half-power beamwidth of the antenna and the physical space constraint of the system. To overcome this, an RSNS using virtual spacing has been proposed in [25]. In this case, the computed RSNS spacings are not physical spacings used by the antenna but instead virtual spacings. A scale factor is computed to translate the element spacing to the virtual spacing for each channel. Since a folding waveform can be generated for a phase difference, $\Delta\phi$, between elements, it can be extended to any spacing of elements

$$d_i = C_i d \tag{3.11}$$

where $d_i$ is the RSNS virtual spacing, $C_i$ is the scale factor and $d$ is the physical element spacing for the single channel.

The threshold voltages can then be found for any spacing provided there is a one-to-one relationship between the phase difference and angle of arrival for the physical spacing of the element. The threshold voltages are computed using the following equation:

$$V_i(\theta) = \cos\left(\frac{C_i kd}{2}\sin(\theta)\right) = \cos\left(\frac{kd_v}{2}\sin(\theta)\right) = \cos\left(\frac{C_i\Delta\phi}{2}\right) \tag{3.12}$$

Thus for any spacing, $d_i$, the output voltage for an angle of arrival can be determined provided the relationship between the phase difference, $\Delta\phi$, and the angle of arrival is unambiguous. This occurs when $d < 1\lambda$ as shown in Figure 17.

For phase combining the number of folding periods, *n*, is given by:

$$n = \frac{d}{\lambda/2} \tag{3.13}$$

However, for amplitude combining, $n = \frac{d}{\lambda}$.

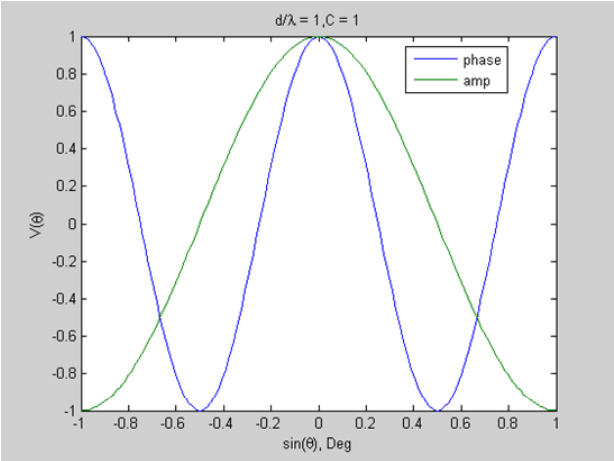



Figure 17. Comparison of amplitude and phase folding waveform (From [26])

### 3. RSNS Implementation with Moduli [5, 9]

In [10], a two-channel RSNS system with virtual spacing was implemented. Moduli [5, 9] were chosen for the system as both values are relatively prime numbers and provide sufficient dynamic range for the purpose of acquiring the signal direction for subsequent tracking. Using Equation (3.6), the dynamic range of the system is evaluated to be 36. For a system that scans $-90^0 \le \theta \le 90^0$, the bin at broadside will be $180^0 / 36^0 = 5^0$.

The number of folds within each modulus is computed from Equation (3.7) and the following number of folds for each virtual channel is calculated

$$n_1 = 36/\big((2)(2)(5)\big) = 1.8 \tag{3.14}$$

$$n_2 = 36/\big((2)(2)(9)\big) = 1 \tag{3.15}$$

For a system operating at 2.4 GHz to achieve the number of folds within the waveform for each channel, using Equation (3.8), the virtual channel spacings are computed as $d_1 = 0.1125$ m and $d_2 = 0.0625$ m.

Since this is an implementation of a virtual RSNS, the scale factor has to be evaluated to determine the threshold voltages. For a physical spacing of 0.065 m, $C_1 = 1.73$ and $C_2 = 0.96$. The prescribed element spacing of 0.065 m will be explained in a later chapter in consideration of the design of the antenna array. The threshold values for the different bins are then computed for the RSNS system, and are shown in Table 2.

Table 2. Voltage threshold values for RSNS using [5, 9] as moduli

| $m/j$ | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|---|---|---|---|---|---|---|---|---|---|
| 5 | -0.5878 | 0 | 0.5878 | 0.9511 | 1 | 0 | 0 | 0 | 0 |
| 9 | -0.866 | -0. 6428 | -0.3420 | 0 | 0.342 | 0.6428 | 0. 866 | 0. 866 | 1 |

To align each folding waveform to the center of the dynamic range for each channel, the phase adjustments, $\varphi_i$, for each channel are found to be -1.2566 and -1.5708 radians. Thus, for a system using quadrature demodulator boards, the measured $I$ and $Q$ data for the antenna elements are used to compute $\Delta\phi$, which can then be used to compute the level of the virtual waveform for the given angle of arrival. Including the phase difference in Equation (3.12), the voltage of the incoming wave can be written as:

$$V_i(\theta) = \cos\left(\frac{C_i \Delta\phi}{2} + \varphi_i\right) \tag{3.16}$$

The computed $V_i$ in Equation (3.16) for both channels are then compared against the RSNS thresholds in Table 2 to determine the direction of the angle of arrival.

## C. DIGITAL BEAMFORMING AND PHASE INTERFEROMETRY

In the tracking techniques described in Section A, several different methods were presented. All the techniques involved the process of receiving the signal from the target,

developing a pointing error signal and then applying it to the beam direction. This involves the process of receiving at a scan angle, developing an error signal, and repositioning the beam at the new scan angle. Digital Beamforming (DBF) represents a slightly different approach to tracking. The field of view of the individual antenna elements in an array is not changing. The computation of the error signal and beam formation and scanning are all done through software[2].

Consider an antenna array aligned along the $x$-axis as shown in Figure 18. As the plane wave arrives at each of the array elements, information on the amplitude and phase are captured by the elements. When this wavefront is not parallel to the array axis, there will be a difference between the received phases at each element. This phase difference, $\Delta\phi$, is proportional to the angle of arrival, $\theta$, frequency, and element spacing, $d$. This phase difference is given by

$$\Delta\phi = \frac{2\pi}{\lambda} d \sin(\theta) \tag{3.17}$$

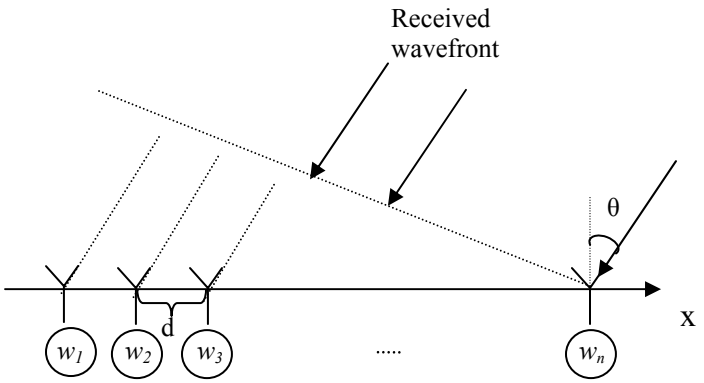



Figure 18. Antenna array using DBF

In DBF, the signals out of the elements are down converted and sampled. The complex data, generally in the form of $I$ and $Q$, are sent to the beamformer computer where the weights are added and the beam formed.

[2] The material presented in this section has been adapted from [21].

The sum beam array factor for the antenna can be expressed by

$$\Sigma \equiv AF_{\Sigma}(\theta,\phi) = \sum_{n=1}^{N} w_n e^{jkd_n \cos\theta} = \sum_{n=1}^{N} A_n e^{j\alpha_n} e^{jkd_n \cos\theta} \tag{3.18}$$

where $N$ is number of elements, $k = 2\pi/\lambda$, $d_n$ is the location of the element relative to the center of the array and $d$ is the distance between the elements:

$$d_n = \frac{\left(2n - \left(N+1\right)\right)}{2} d \tag{3.19}$$

A complex weight $w_n = A_n e^{j\alpha_n}$ added to each array element.

When the IQ demodulators are used in the receive branch of the antenna array, the output of the corresponding demodulators will be based on the following equations:

$$I_n = \cos\left(kd_n \cos\theta\right) \tag{3.20}$$

$$Q_n = \sin\left(kd_n \cos\theta\right) \tag{3.21}$$

If an equal amplitude ($A_n$ = 1) weight is applied to each array element based on the scan direction, $\theta_s$,

$$w_n = e^{-jkd_n \sin\theta_s} \tag{3.22}$$

The sum pattern is obtained from the evaluation of Equation (3.18), while the difference pattern is evaluated using,

$$\Delta \equiv AF_{\Delta}(\theta,\phi) = \sum_{n=1}^{N/2} A_n e^{j\alpha_n} e^{jkd_n \cos\theta} - \sum_{N/2}^{N} A_n e^{j\alpha_n} e^{jkd_n \cos\theta} \tag{3.23}$$

Thus, using the sum and difference beams from the output of the antenna array, a plot similar to Figure 16 can be obtained and the phase error of the direction to the target can be evaluated using Equation (3.1). For the purpose of monopulse tracking, it is only applicable when operating within the linear region of Figure 16.

In this chapter, the fundamentals of tracking principles were discussed. In the next chapter, the receiver architecture will be discussed.

# IV. RECEIVER ARCHITECTURE

This chapter discusses the direct-conversion receiver architecture, the mathematical derivation of the *I*/*Q* frequency downconversion process and the three main problems related to direct conversion.

## A. DIRECT CONVERSION

The ubiquitous superheterodyne architecture can be found in almost all receivers. However, this architecture is not without its own difficulties. The superheterodyne architecture requires two or more stages of detecting and filtering, i.e., converting the carrier frequency to an intermediate frequency (IF) first, before extracting the modulation. In comparison, the direct-conversion receiver has a much simpler architecture as only a single conversion stage is required to perform direct demodulation of an RF modulated carrier to baseband frequencies, where the signal can be directly detected and the conveyed information recovered. Hence, it has lower complexity and power consumption. The reduced component count that results from eliminating intermediate frequency stages also means the direct-conversion architecture intrinsically costs less. Also, sampling at baseband allows lower sampling frequency and a lower input (analog) bandwidth. Another important benefit of eliminating the intermediate frequency stages are that the troublesome problem of image rejection can be ignored.

Mathematically, the mixing process gives the output voltage

$$\begin{aligned} V_{out} &= \cos\left(\omega_{RF}t\right)\cos\left(\omega_{LO}t\right) \\ &= \frac{1}{2}\left[\cos\left(\omega_{RF}-\omega_{LO}\right)t+\cos\left(\omega_{RF}+\omega_{LO}\right)t\right] \end{aligned} \tag{4.1}$$

The mixing process will produce a pair of frequencies—sum and difference frequencies. Since the direct-conversion receiver down converts directly to zero-

IF $(\omega_{RF} = \omega_{LO})$, no image frequency exists that will yield a mixer output at the desired baseband frequency. Figure 19 illustrates the mixing process and the sum and difference frequencies generated by the mixer.

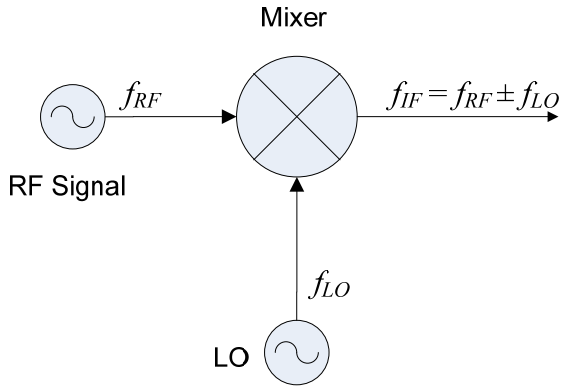



Figure 19. The generation of the sum and difference frequencies by the mixing process

Since *I* and *Q* data can be used to represent any changes in magnitude and phase of a message signal, quadrature *I* and *Q* channels are necessary in typical phase and frequency modulated signals to separate the two sidebands. The two sidebands of the RF spectrum contain different information for phase and frequency modulated signals and irreversible corruption results if they are allowed to overlap each other without being separated. Similar to the benefits of *I*/*Q* modulation, it is also very expensive and difficult to design and build hardware circuits to directly recover the phase of a RF carrier sine wave [27]. Hence, another benefit of using quadrature detection is that it is much simpler in the hardware circuit to induce a 90-degree phase shift between the carrier signals used for the *I* and *Q* mixers to recover the voltage amplitudes of separate *I* and *Q* input signals. The output *I* and *Q* components form the complex baseband signal.

For quadrature down conversion, either the RF or LO signal must be shifted by 90 degrees. This 90-degree phase shift provides the mechanism to distinguish the *I* and *Q* components of the RF signal [28]. Since shifting the RF signal generally entails severe noise-power-gain tradeoffs, it is desirable to perform the phase shift on the LO signal.

Figure 20 shows the two sidebands after down conversion to zero-IF, where $f_{LO} = f_{RF} \equiv f_O$.

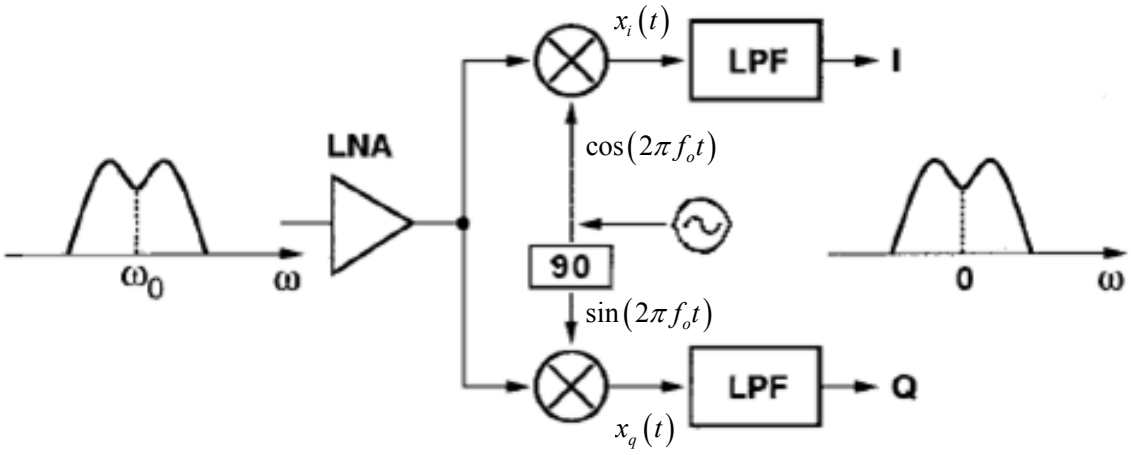



Figure 20. Direct-conversion architecture (After [29])

The in-phase baseband component can be described by:

$$x_i = \mathrm{Re}\left[ A_c e^{-j\left(2\pi f_o t + \theta(t)_{FM}\right)} \right] \cos\left(2\pi f_o t\right) \tag{4.2}$$

where $\theta(t)_{FM}$ is the frequency modulated message signal. Since $\mathrm{Re}[x] = \frac{x + x^*}{2}$, where $x$ is a complex number, and a cosine and sine wave can be expressed in the form of Euler's identities as:

$$\cos\left(2\pi f_o t\right) = \frac{e^{j2\pi f_o t}}{2} + \frac{e^{-j2\pi f_o t}}{2} \tag{4.3}$$

$$\sin\left(2\pi f_o t\right) = \frac{je^{-j2\pi f_o t}}{2} - \frac{je^{j2\pi f_o t}}{2} \tag{4.4}$$

where $e^{jx}$ represents the positive frequency component and, $e^{-jx}$ the negative frequency component. Hence, Equation (4.2) develops to:

$$x_i = \frac{A_c\left(e^{j\left(2\pi f_o t+\theta(t)_{FM}\right)}+e^{-j\left(2\pi f_o t+\theta(t)_{FM}\right)}\right)}{2}\left(\frac{e^{j2\pi f_o t}+e^{-j2\pi f_o t}}{2}\right)$$
$$= \frac{A_c}{4}\left(e^{j\left(4\pi f_o t+\theta(t)_{FM}\right)}+e^{j\theta(t)_{FM}}+e^{-j\theta(t)_{FM}}+e^{-j\left(4\pi f_o t+\theta(t)_{FM}\right)}\right)$$
$$= \frac{A_c}{2}\left(\mathrm{Re}\left[e^{j\left(4\pi f_o t+\theta(t)_{FM}\right)}\right]+\mathrm{Re}\left[e^{j\theta(t)_{FM}}\right]\right) \qquad (4.5)$$

After low pass filtering, the high frequency disappears, leaving behind the low pass frequency spectrum at *I* in Figure 20. Hence, Equation (4.5) becomes

$$I = \frac{A_c}{2}\mathrm{Re}\left[e^{j\theta(t)_{FM}}\right] \qquad (4.6)$$

The mathematical derivation from Equation (4.5) is illustrated in the frequency domain by Figure 21. It essentially describes what happens when a bandpass signal is mixed with $\cos\left(2\pi f_o t\right)$ in the upper or *I* channel path in Figure 20. The single arrow lines show the multiplication with the positive frequency component and the double dot-dashed lines show the multiplication with the negative frequency component. The shaded area highlights the low pass filtered spectrum.

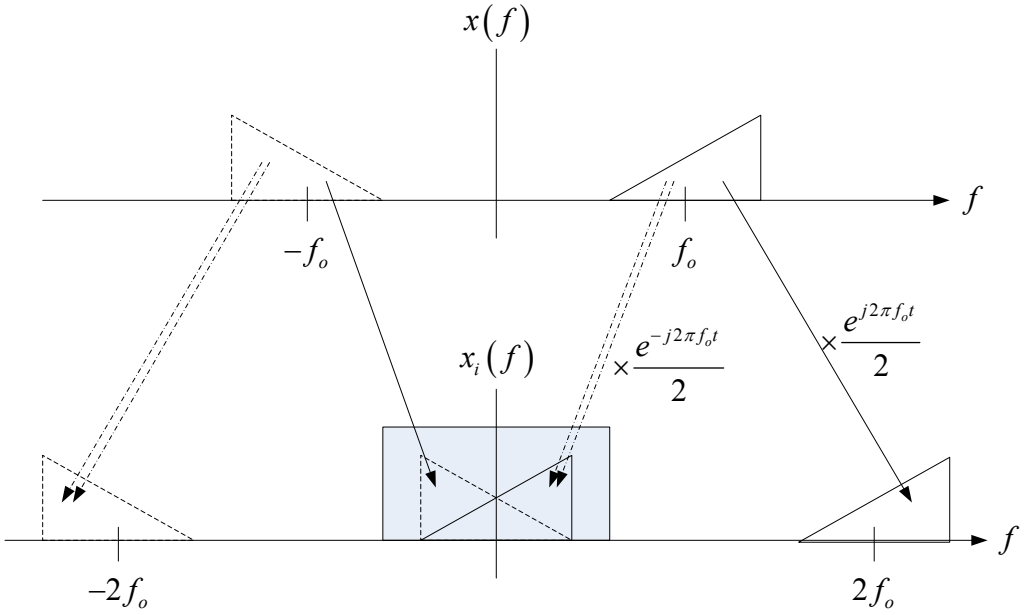



Figure 21. Graphical demonstration of mixing in the *I* channel

Similar to the in-phase baseband component, using the identity $\mathrm{Im}[x]=\frac{x-x^*}{2}$ and Equation (4.4), the quadrature baseband component can be shown as follows:

$$\begin{aligned} x_q &= \frac{A_c\left(e^{j\left(2\pi f_o t+\theta(t)_{FM}\right)}-e^{-j\left(2\pi f_o t+\theta(t)_{FM}\right)}\right)}{2}\left(\frac{je^{-j2\pi f_o t}-je^{j2\pi f_o t}}{2}\right) \\ &= \frac{A_c}{4}\left(je^{j\theta(t)_{FM}}-je^{j\left(4\pi f_o t+\theta(t)_{FM}\right)}-e^{-j\left(4\pi f_o t+\theta(t)_{FM}\right)}+je^{-j\theta(t)_{FM}}\right) \\ &= j\frac{A_c}{4}\left(\mathrm{Im}\left[e^{j\left(4\pi f_o t+\theta(t)_{FM}\right)}\right]-\mathrm{Im}\left[e^{j\theta(t)_{FM}}\right]\right) \end{aligned} \tag{4.7}$$

Again, after low pass filtering, the low pass frequency spectrum of *Q* is given by:

$$Q = j\frac{A_c}{2}\mathrm{Im}\left[e^{j\theta(t)_{FM}}\right] \tag{4.8}$$

This derivation is illustrated in Figure 22, when a bandpass signal is mixed with $\sin\left(2\pi f_o t\right)$ in the lower or *Q* channel path in Figure 20. Figure 23 shows the final output signal (with two sidebands) after down conversion and summation of the *I* and *Q* channels. The amplitude of the final signal is two times that of the single sideband of the bandpass signal.

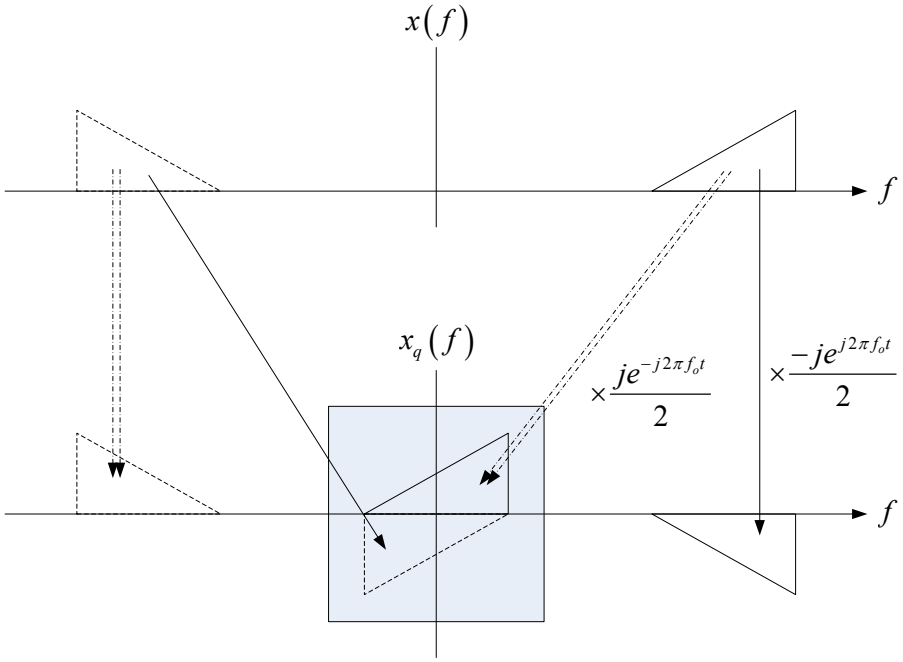



Figure 22. Graphical demonstration of mixing in *Q* channel

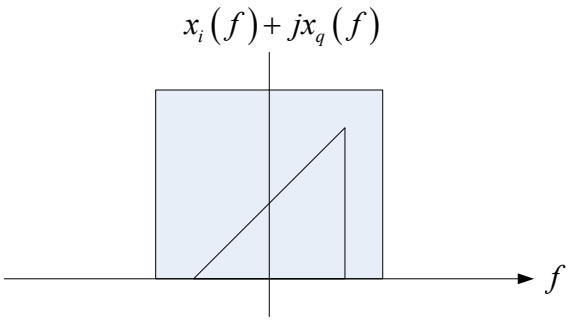



Figure 23. Final output with amplitude doubled

## B. PROBLEMS ASSOCIATED WITH THE DIRECT CONVERSION ARCHITECTURE

The direct-conversion architecture is not without drawbacks. There are three main problems associated with the direct-conversion architecture discussed in the following sections. Hence, practical designs using direct conversion need to consider problems associated with I/Q mismatches, dc offsets and second order distortions.

### 1. I/Q Mismatch

Ideally, the in-phase and quadrature channels carry orthogonal signals with no correlation to each other. However, due to mismatches in gain and phase in the two channels, correlation will occur that corrupts the original information and makes it more difficult to recover. Phase mismatch error occurs due to the unequal path lengths in the *I*/*Q* channels. Another source of mismatch arises from inaccuracy in the LO quadrature generator. This corrupts the down-converted signal. However, this problem can be circumvented with higher levels of integration [29].

### 2. DC Offset

The direct-conversion receiver suffers from self mixing of the LO due to LO leakage, also commonly known as feedthrough of the LO signal. Self-mixing of the LO occurs because of parasitic LO-RF coupling, or from self-mixing of the RF signal due to RF-LO coupling shown in Figure 24. This causes a dc component at the mixer output. Since the RF or LO signal is in-band, it cannot be filtered out. One way to solve this is to pay particular attention to the layout of the printed circuit board (PCB) so that there is an adequate amount of isolation between the RF and LO inputs and also to ensure that the minimum amount required for shielding is available [30]. Mismatches in the dc components of the signals at the modulator for the transmitted signal and also mismatches in the *I*/*Q* demodulator itself can also cause dc offsets. In this architecture, since the down-converted signal extends to zero-IF, extraneous offset voltages can corrupt and saturate the subsequent stages.

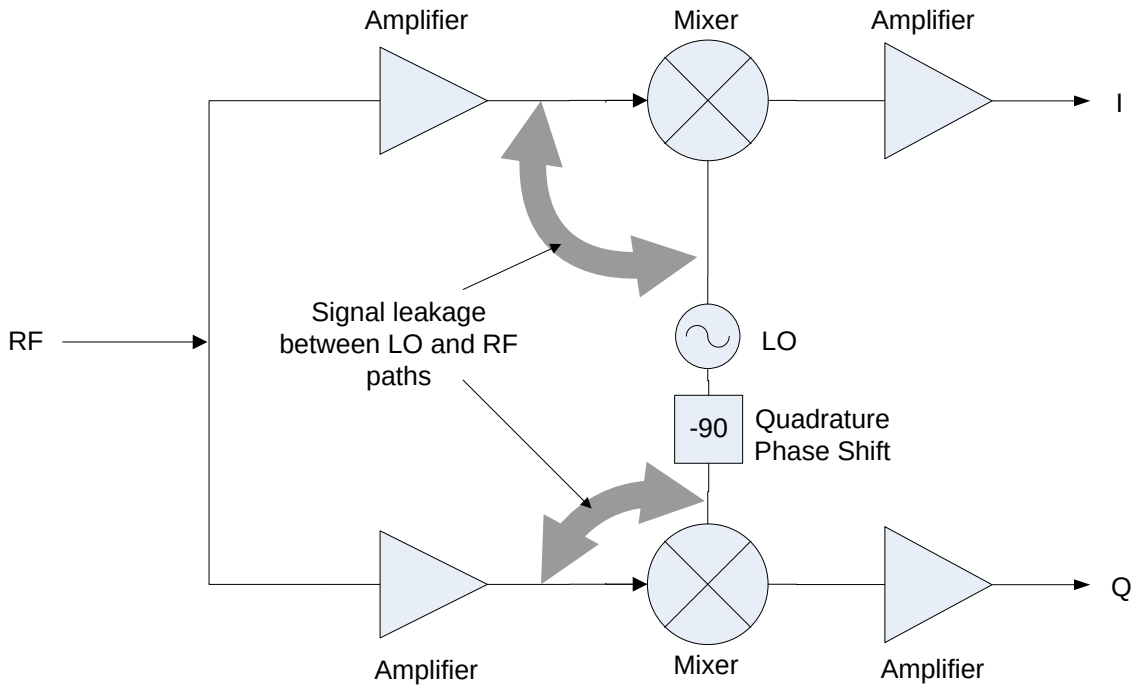



Figure 24. Diagram to show RF-LO leakage

### 3. Second Order Distortion

Second order distortion contributes a baseband term of the form $\cos\left[\left(\omega_{RF}-\omega_{LO}\right)t\right]$, which is likely to be in-band for a direct conversion design. Another form of second-order non-linearity is the second harmonic of the desired RF signal, which is down converted to the baseband if mixed with the second harmonic of the LO output. In actual applications, second order distortion can become problematic if there is a single strong interferer nearby. These interferers are then more likely to produce unwanted products that fall directly on top of the desired baseband signal, which cannot be filtered out. This is illustrated in Figure 25. Specifically, second order distortion due to a tone interferer will give rise to a dc offset at the mixer output. A modulated signal due to second order nonlinearity will appear at the baseband output if the interferer is modulated [31]. The second order intercept point (IP2) is a measure of second order non-linearity.

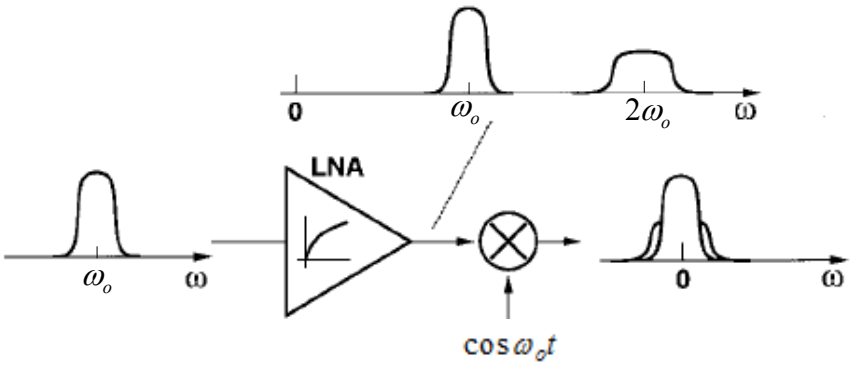



Figure 25. Down-conversion of signal harmonics (After [32])

This chapter discussed the direct-conversion receiver architecture, its benefits and associated problems, and mathematically derived the *I*/*Q* frequency downconversion process. The next chapter overviews both the hardware and software system design and provides an analysis of the system performance.

THIS PAGE INTENTIONALLY LEFT BLANK

# V. SYSTEM DESIGN OVERVIEW

In this chapter, the hardware and software components of the system are discussed in detail. An analysis of the system performance is also provided.

## A. SYSTEM OVERVIEW

The selected receiver architecture is a direct-conversion receiver, also known as homodyne, or, zero-IF receiver. After the low noise front end, the desired carrier frequency is down converted to baseband using an *I*/*Q* demodulator. As seen in Figure 26, each receiver channel (path) is primarily made up of five main blocks: the antenna element, the down-converter, the local oscillator (LO), the analog-to-digital converter (ADC) and processor blocks. The current array design has six channels. The first four blocks of the system are implemented in hardware while the last block is implemented in software. In this chapter, the hardware and software components of the system are discussed in detail.

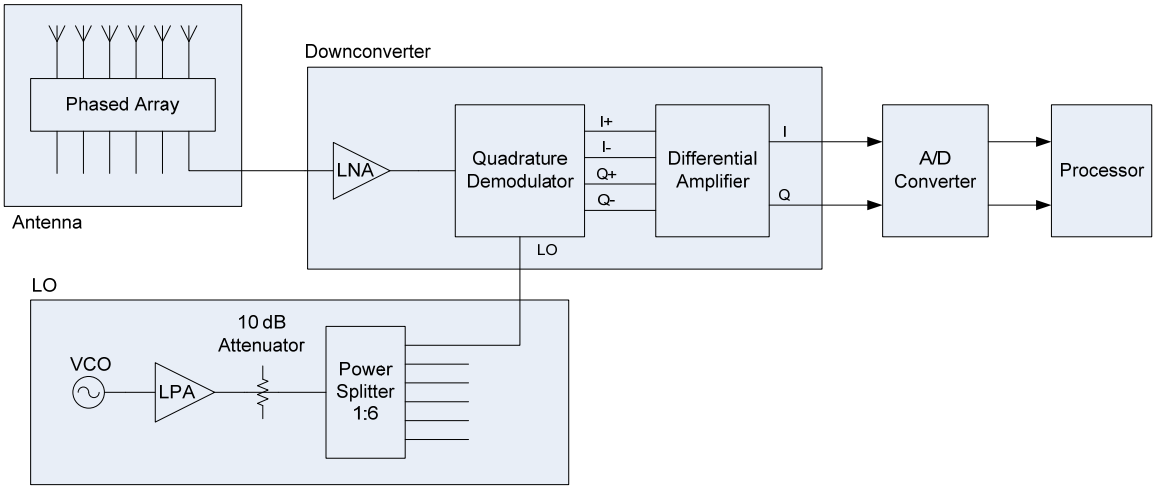



Figure 26. System block diagram

## B. HARDWARE OVERVIEW

The receiver demodulates the incoming signal by mixing it with a local oscillator signal that is the same frequency as the carrier signal. From the mixing process, sum and difference frequencies are generated at the baseband *I*/*Q* output ports. For the zero-IF scenario, the difference frequency reveals the baseband envelope of the wanted signal (i.e., the wanted modulation signal is obtained immediately with just one stage of down-conversion). The modulation signal is then passed to the ADC after filtering through a low-pass filter (LPF) network to reject unwanted higher frequency components (the sum frequency components generated by the mixing process).

Prior to the ADC, the magnitude of the filtered baseband I/Q signal is amplified to allow the *I*/*Q* signal levels to be adjusted to an optimum level for ADC conversion. In general, additional filtering is applied before the ADCs to ensure that high-frequency noise and potential leakage or interfering tones do not alias back into the desired signal bandwidth.

### 1. Antenna

The antenna array design is a linear array with six two-element subarrays (Figure 27). The array axis is the azimuth plane.

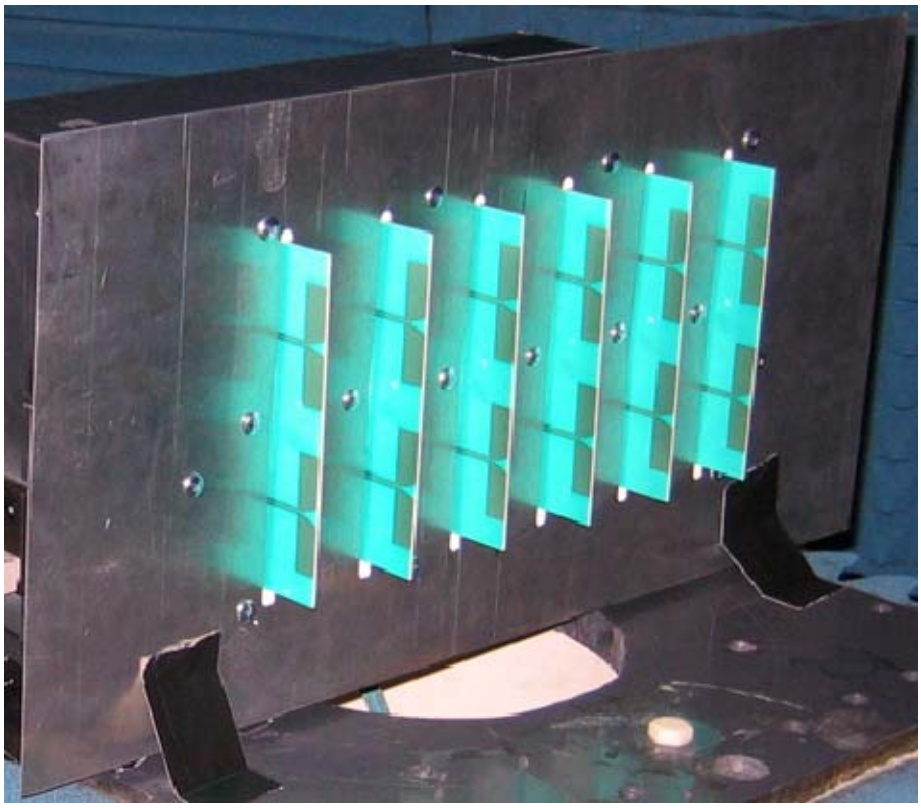


Figure 27. Picture of the antenna array

Each single subarray is comprised of two half-wave dipole antennas stacked in the elevation plane for increased directivity. Two elements also provide a higher signal level at the input to the demodulator board, allowing for longer range operation. The dipole arms are fed by power splitters as shown in Figure 28. One of the arms of the splitter is $\lambda/2$ longer than the other to create a $180^\circ$ phase shift so that the currents on the two arms are in phase with each other. The $180^\circ$ compensation is flipped for one of the two dipoles. This is because the rat-race hybrid is fed from the difference rather than the sum to avoid a feed-through connector or crossover on the PCB. Each two-element subarray is fabricated from Rogers RO4003C substrates with relative dielectric constant $\varepsilon_r = 3.38$ and thickness $d = 60\,\text{mils}\,(1.52\text{ mm})$.

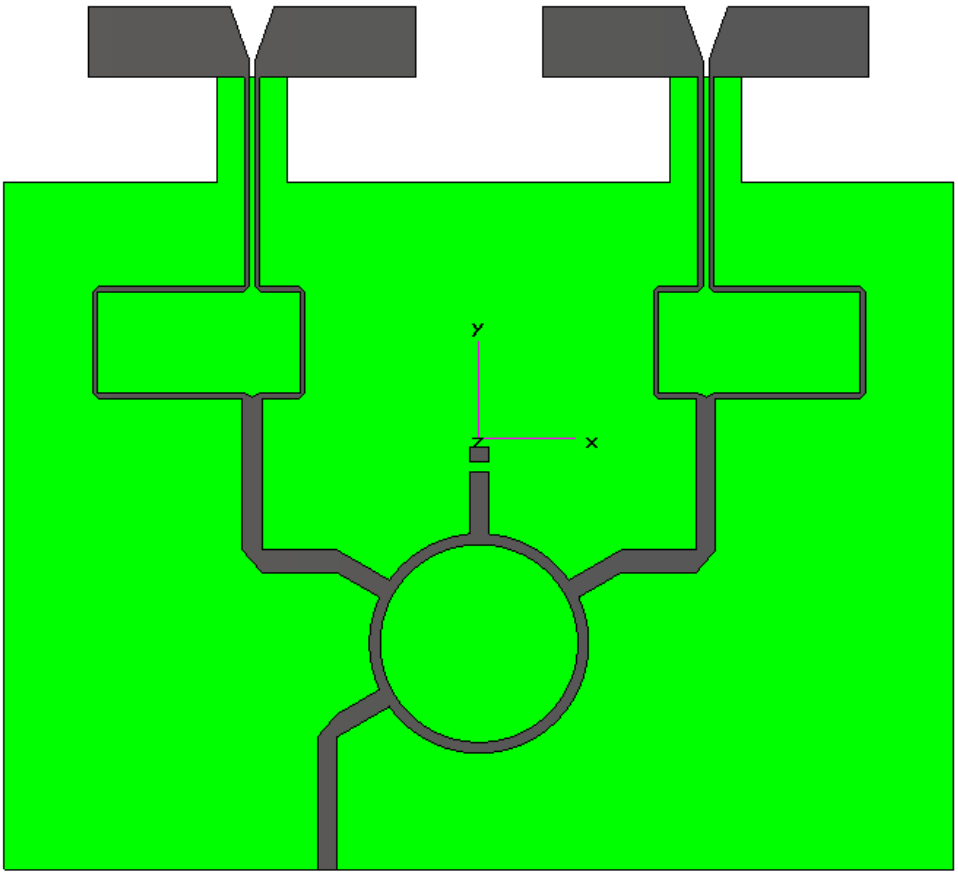



Figure 28. PCB layout of the two element subarray

The width, $W$, of the microstrip lines depends on the characteristic impedance, $Z_0$, and relative dielectric constant, $\varepsilon_r$, of the substrate used. A Matlab program was used to calculate the widths of the microstrip lines from [33]:

$$\frac{W}{d}=\begin{cases}\dfrac{8e^{A}}{e^{2A}-2} & \dfrac{W}{d}<2\\[2ex] \dfrac{2}{\pi}\left[B-1-\ln\left(2B-1\right)+\dfrac{\varepsilon_r-1}{2\varepsilon_r}\left(\ln\left(B-1\right)+0.39-\dfrac{0.61}{\varepsilon_r}\right)\right] & \dfrac{W}{d}>2\end{cases} \tag{5.1}$$

where

$$A=\frac{Z_0}{60}\left[\sqrt{\frac{\varepsilon_r+1}{2}}+\frac{\varepsilon_r-1}{\varepsilon_r+1}\left(0.23-\frac{0.11}{\varepsilon_r}\right)\right] \tag{5.2}$$

$$B = \frac{377\pi}{2Z_0\sqrt{\varepsilon_r}} \tag{5.3}$$

For the lines feeding the rat-race hybrid in Figure 28 have $Z_0 = 50\,\Omega$ so $W/d = 2.32$ and the width is calculated to be $W = 139\,\text{mils} = 3.53\,\text{mm}$. Since equal power from the rat-race hybrid is desired, the computed characteristic impedance is $Z_{rr} = \sqrt{2}Z_0 = 70.8\,\Omega$. As $W/d = 1.27 < 2$, the line width of the rat-race's hybrid itself is calculated to be 76.3 mils = 1.94 mm. Going through a similar process, the trace width of the $180^\circ$ splitter/coupler is calculated to be 35.7 mils = 0.91 mm from a characteristic impedance of $Z_s = 100\,\Omega$. This is close to the impedance at the feed point of the dipole arms.

### 2. Down-converter

The down-converter block in Figure 26 is composed of three main parts: (1) the low-noise amplifier (LNA), (2) the quadrature demodulator and (3) a differential amplifier stage.

#### *a. Low-Noise Amplifier (LNA)*

The LNA is a connectorized component from RF Bay Inc. (LNA-2700) that provides a gain of 25 dB and an output 1 dB compression point (OP1dB) of 10 dBm. It also has a noise figure (NF) of approximately 1.7 dB. The purpose of the LNA is to provide gain at the front end of the receiver so that weaker signals can be detected. Also, another important purpose is to "fix" the noise figure of the receiver. For a cascaded system, the overall noise factor can be calculated from:

$$F_{overall} = F_1 + \frac{F_2 - 1}{G_1} + \frac{F_3 - 1}{G_1 G_2} + \cdots + \frac{F_n - 1}{G_1 G_2 \cdots G_n} \tag{5.4}$$

where $F_n$ and $G_n$ are the noise figure and gain of the $n^{th}$ component, respectively. The noise figure in dB is simply given:

$$NF = 10 \log F \tag{5.5}$$

A low noise figure is desirable because it enables the receiver to have superior sensitivity. Hence, we can see that an amplifier stage with high gain and low noise at the front can greatly influence the overall noise figure of the receiver. Therefore, the subsequent components do not need to have low noise figures, which translate to lower costs as low noise components inherently cost more because they are more difficult to manufacture and design.

### *b. Quadrature Demodulator*

The quadrature demodulator is based on the Analog Devices AD8347 Direct Conversion Quadrature Demodulator integrated circuit (IC). As its names implies, the AD8347 is designed for quadrature demodulation directly to baseband frequencies, where it is convenient for connection directly to A/D converters. A graphical representation is shown in Figure 29. It consists of a phase splitter, a pair of identical low-pass filters and Gilbert-cell mixers with three stages of variable gain amplifiers providing a total of 69.5 dB of automatic gain control (AGC).

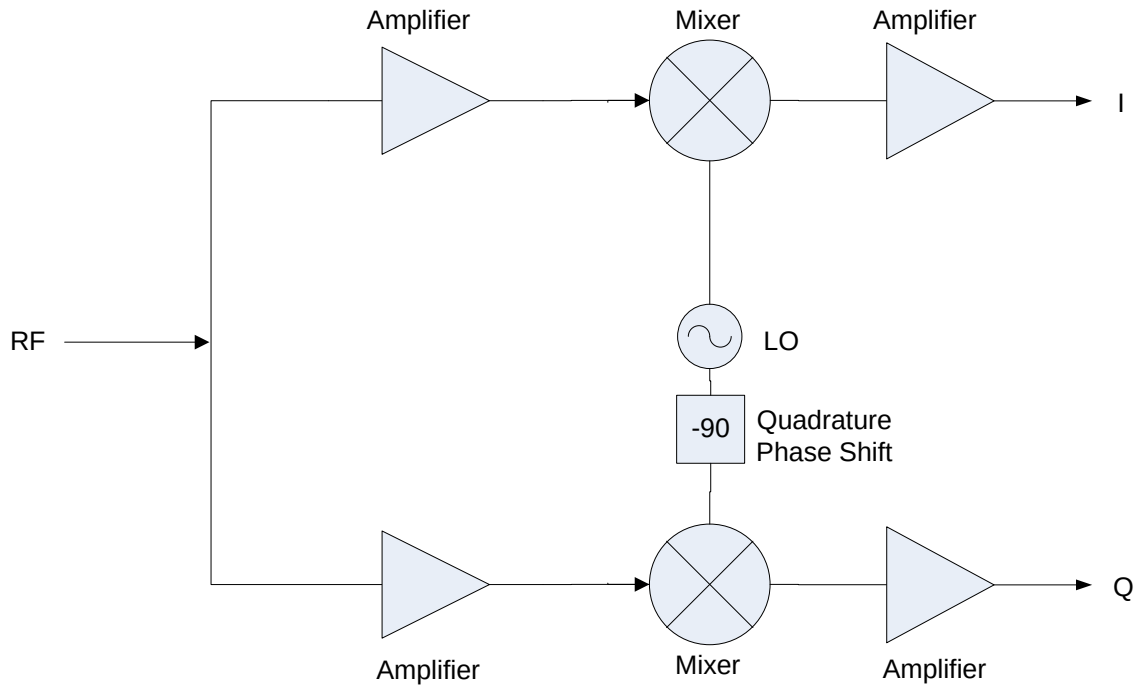



Figure 29. Graphical representation of the quadrature demodulator

As mentioned in Chapter IV, the mixing process essentially generates two frequencies, i.e., the sum and difference frequency terms as given in Equation (4.1). Since $\omega_{RF} = \omega_{LO}$ in homodyne receivers, Equation (4.1) reduces to:

$$V_{out} = \frac{A(t)}{2}\left[1 + \cos\left(2\omega_{RF}\right)t\right] \tag{5.6}$$

After the mixing process, ignoring the factor of $1/2$, the first term is a signal at zero-IF, and the second term at twice the carrier frequency. The signal at twice the carrier frequency will be rejected by the LPF. The wanted signal at zero-IF is retained and it is proportional to the complex envelope $I(t) + jQ(t)$ [32].

### *c. Differential Amplifier*

The illustration in Figure 30 shows one-half of the differential amplifier block. This block can either be the *I* or *Q* channel. The other half of the block is identical. It is built on an operational amplifier (Op-Amp) chip from National

Semiconductor (LM622), which has four wideband video op-amps. The differential amplifier block is designed to satisfy three tasks as follows.

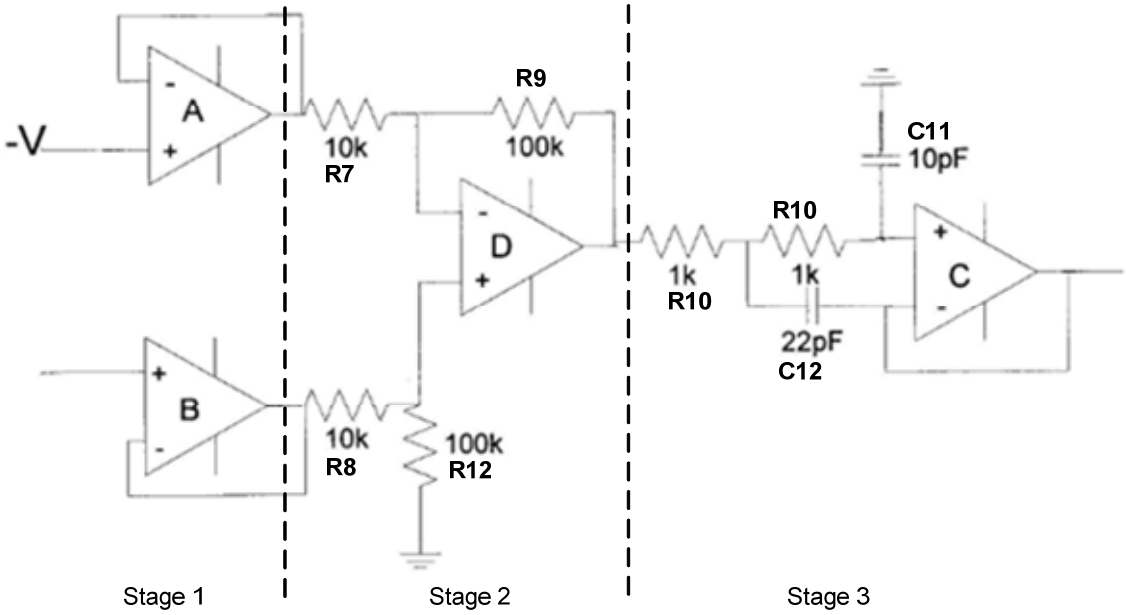



Figure 30. Schematic diagram showing one-half of the differential amplifier block

(1) Reduce cost and equipment complexity. The AD8347 Direct Conversion Quadrature Demodulator IC produces four differential outputs (IOPP, UOPN, QOPP, QOPN), thus allowing the device to be operated with single-sided dc power supply. However, this also means that it requires four ADCs in the digital processor section of the receiver. The first stage of the block converts the four differential input channels to two output channels, thus reducing the required number of ADCs (equipment complexity). It also reduces the data processing requirements.

(2) Provide voltage gain. The second stage of this block provides a voltage gain of 10. This helps to provide additional amplification to the baseband signal. The voltage gain ( $A_v$ ) of the second stage can be specified by changing the values of the resistors $R9/R12$ and $R7/R8$. The relationship between $R9/R12$ and $R7/R8$ is described by:

$$A_v = \frac{R9}{R7} = \frac{R12}{R8} \tag{5.7}$$

In decibels (dB), the gain can be calculated by:

$$G(dB) = 20\log A_v \tag{5.8}$$

(3) Low pass filtering. The final stage of the block acts as an anti-aliasing filter for the receiver. The LPF also helps to eliminate higher order unwanted frequency components that are produced by the demodulation process. The 3 dB cutoff frequency is 10 MHz. This is set by the NTSC video signal, which has a bandwidth of 6 MHz [34]. The 3 dB cutoff frequency ( $f_c$ ) can be calculated using:

$$f_c = \frac{1}{2\pi\sqrt{R10\,R11\,C11\,C12}} \tag{5.9}$$

### 3. Local Oscillator (LO)

The LO is used to provide a signal source for frequency down conversion.

#### *a. Voltage-Controlled Oscillator (VCO)*

An oscillator is a non-linear device that converts DC power to a sinusoidal-steady AC waveform [33]. It is the primary device in the LO branch. For this project, the oscillator chosen is a VCO. A VCO, as its name implies, has an oscillator that provides a signal whose frequency is controllable using an analog voltage signal. Having a controllable output frequency is an essential function of this VCO. The controllable output frequency is required because the VCO must be fine-tuned to provide the LO frequency to match that of the received signal for the homodyning process.

#### *b. Low-Power Amplifier (LPA)*

The aim of the LPA in the LO branch is to provide a power gain to the LO signal level such that the LO signal power level is sufficient to drive the mixer. Studies have shown that insufficient LO drive power levels will lead to high conversion loss and high levels in the intermodulation products [35]. These will in turn degrade the performance of the mixer and demodulation process and introduce phase errors.

However, the LO signal level cannot be too high. This is because the LO drive circuit would require immense amount of prime power. This on its own would create other problems in terms of electromagnetic interference (EMI). In addition, a lower LO input power also results in less leakage and consequently less dc offset problems.

#### *c. Attenuator*

The attenuator is added to pad the LO power level to ensure that an optimum drive level of around -8 dBm is attained.

#### *d. Power Splitter*

The six-way power splitter divides the LO power from a single source to the six quadrature demodulators. It is specified to have a maximum of 8 degrees of phase difference between ports. Together with the employment of phase matched cables, the phase differences between the six LO paths should be minimized.

### 4. A/D Converter

The ADC selected here is the National Instruments (NI) PXI-5112 high-speed digitizer. The 5112 is designed to have deep onboard acquisition memory, wide analog bandwidth, and a large analog input range. Because the NI 5112 is based on the high-speed PCI bus, acquiring and processing large waveforms is much faster than with comparable GPIB-based instruments.

The ADC samples and converts the analog input signals to digital values up to a rate of $100\,\mathrm{MS/s}$ (Mega samples per second). The sampling rate must satisfy Nyquist's condition, which states that a signal must be sampled at least twice as fast as the highest frequency component of interest. This means the Nyquist frequency is the sampling rate divided by two $\left(f_s/2\right)$, and for the 5112 to acquire a signal without aliasing, the signal frequency must be below the Nyquist rate of $50\,\mathrm{MHz}$.

The 8-bit resolution also lowers quantization noise and increases sensitivity over other lower resolution ADCs. Also, since the number of quantization levels is $2^N$, where $N$ is the number of bits, a resolution of 8 bits can encode an analog input to one in 256 different levels. Depending on the application, the values can represent the ranges from 0 to 255 (i.e., an unsigned integer) or from -128 to 127 (i.e., a signed integer). The 5112 has an adjustable voltage range so that the 256 levels need not be spread over the maximum range of $\pm 25\,\mathrm{V}$.

## C. SOFTWARE OVERVIEW

In developing an antenna array for the purpose of receiving data from a UAV, the proposed system should be able to track the UAV, retrieve the message signal from the transmitted signal and then finally display the image or video on an output terminal. Since most processing is done in software, it was found to be more prudent to develop the system in a modular approach. This allows each module to be developed and tested separately so as to facilitate eventual integration of the various hardware and software components. In this section, the development of three modules will be discussed: 1) NTSC Decoding, 2) FM Demodulation and 3) Tracking algorithm. The following sections cover the implementation of the various principles for each module that had been discussed in Chapters II and III. Several of the modules were built as a proof of concept and require additional modifications before they can be used in the final software solution. Each software module was tested separately and the results are included in Chapter VI.

### 1. NTSC Decoding Module

In the software architecture described for the proposed solution, one key feature that had to be developed is a software module capable of retrieving the video signal embedded in the received signal at the antenna. In Chapter II, the different video signal formats were discussed. For the solution presented in this thesis, the software developed used the NTSC format (mono). Table 3 shows a summary of the information presented in Chapter II on the NTSC format.

Table 3. Summary of NTSC parameters used in software development

| NTSC Format | |
|---|---|
| Vertical Line Resolution | 525 (485 for data, 40 for Sync Pulses) |
| Frame Rate | 30 frames/sec |
| Vertical Line Rate (mono) | 15,750 Hz |
| Active Video Duration | 52.66 μsec |
| Pixels / Line | 640 |
| Image Size | 640 x 480 pixels |
| Pixel Rate | 12.15x106 pixels/sec |

Figure 31 shows the generic flow chart of the NTSC decoding module developed for a single channel. There are three separate sub VIs that were used to develop this program, *Open and Fetch from 5112*, *Fetch from 5112* and *NTSC Processor*. These sub VIs were developed by Professor David C. Jenn and modified for use in the module presented here.

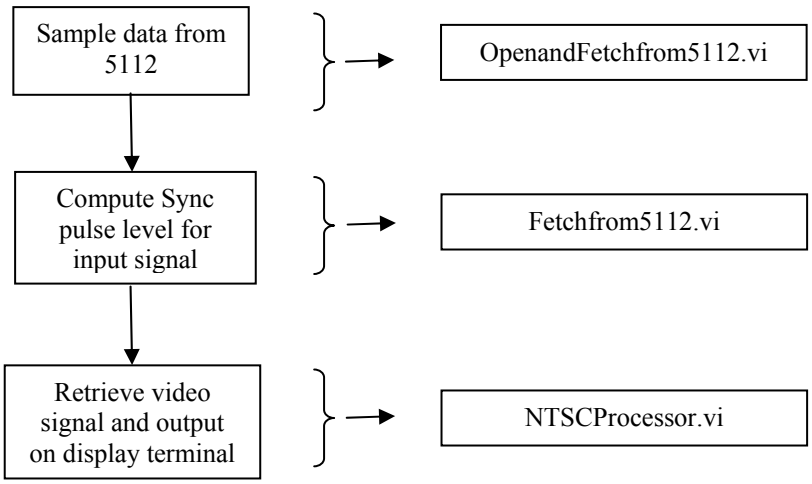



Figure 31. Software flow diagram for NTSC decoding module and the associated sub VIs

Figure 32 shows the NTSC decoding module. As an input, the program takes the configuration settings for sampling of the 5112 boards as well as the settings for the NTSC decoding, and generates a video output image on the display. The program first calls the *Fetch from 5112* sub VI, which samples the 5112 to obtain the data of the incoming signal in terms of *I* and *Q* components. This VI internally calls *Open and Fetch from 5112* to obtain the data from the boards. The Vsync level is computed within this sub VI before it is sent together with the digitally sampled data to the *NTSC Processor* sub VI, where the video signal is recovered from the input signal, generating the output video image. The sub VIs are explained in the following sections.

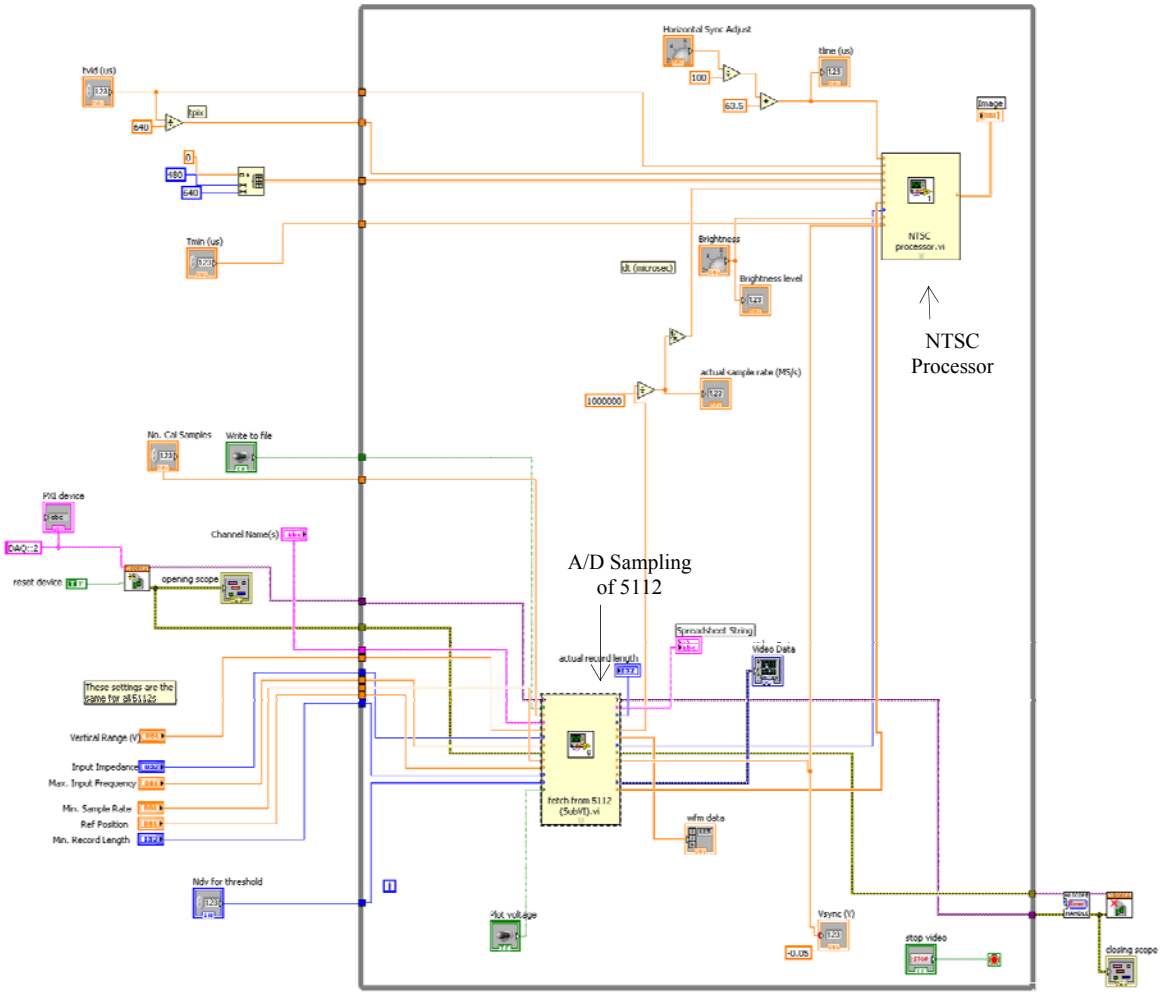



Figure 32. NTSC Decoding module

### *a. Open and Fetch From 5112 Sub VI*

Figure 33 shows the *Open and Fetch from 5112* sub VI. This VI is called from *Fetch 5112* sub VI. It primarily uses the sampling configuration settings to set up and obtain the data from the 5112 board. Data obtained from channel 0 and channel 1 (*I* and *Q* components, respectively) are returned. The trigger coupling is set to dc while the trigger source is set to 'Immediate' to ensure continuous acquisition of data. The other settings used required for this sub VI are shown in Table 4.

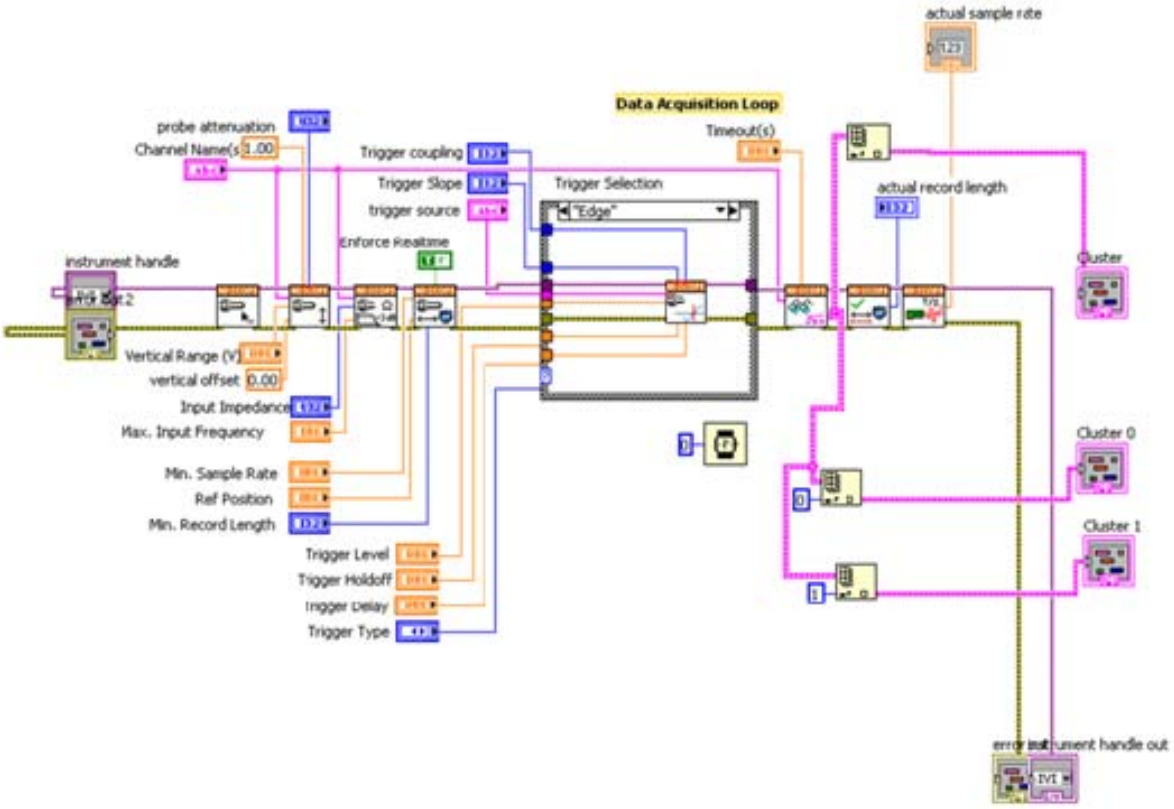



Figure 33. Open and Fetch from 5112 sub VI

Table 4. Configuration settings for Open and Fetch from 5112 sub VI

| Configuration Settings | |
|---|---|
| Min Sample Rate | $20 \times 10^6$ samples / sec |
| Min Record Length | 150, 000 |
| Vertical Range | 2 V |
| Trigger Coupling | dc |

### *b. Fetch From 5112 Sub VI*

The sub VI that is called directly from the NTSC decoding module for data acquisition is the *Fetch from 5112* sub VI. This VI in turn calls the *Open and Fetch from 5112* sub VI, which has been explained in the preceding section. Once the data has been obtained from the 5112, *Fetch From 5112* runs a calibration subroutine to obtain the minimum voltage value, which is subsequently used to set the threshold for Vsync. This information is used in the *NTSC Processor* VI to determine the presence of a new field. The significance of this variable will be explained in the section on the *NTSC Processor*. Figure 34 and Figure 35 show the *Fetch From 5112* sub VI.

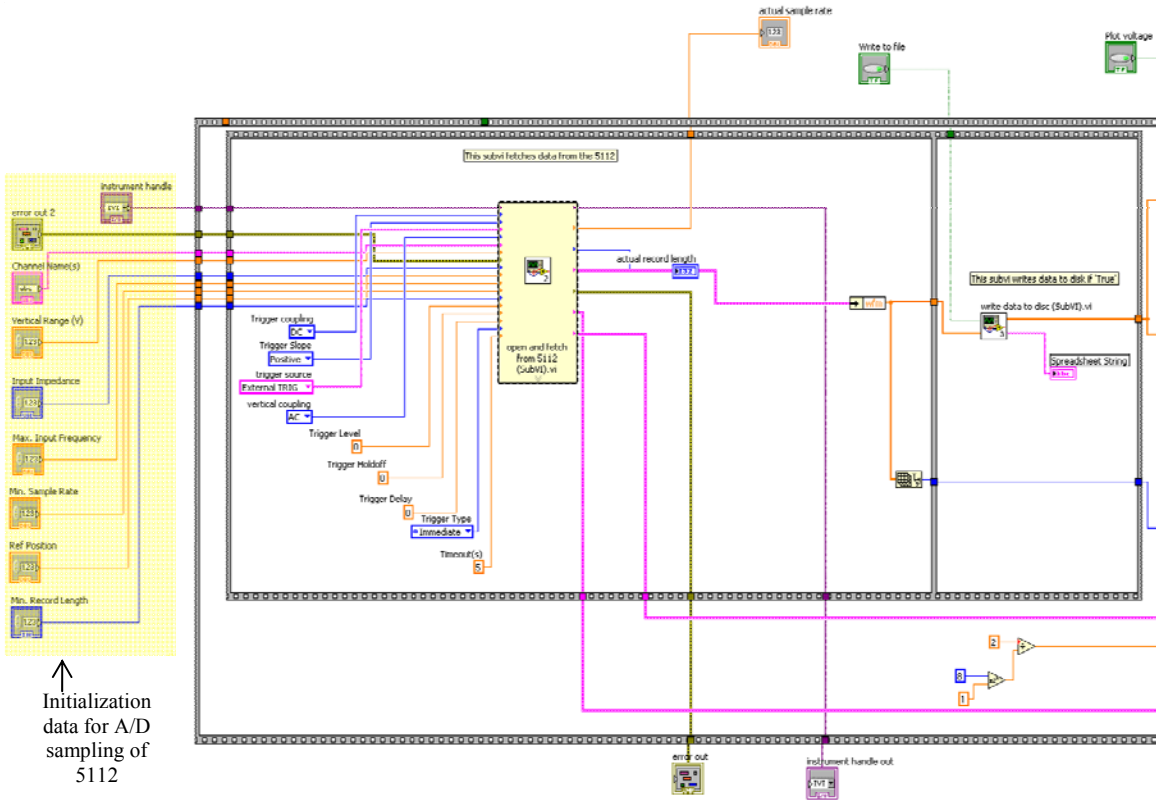



Figure 34. *Fetch From 5112* sub VI - Data Acquisition

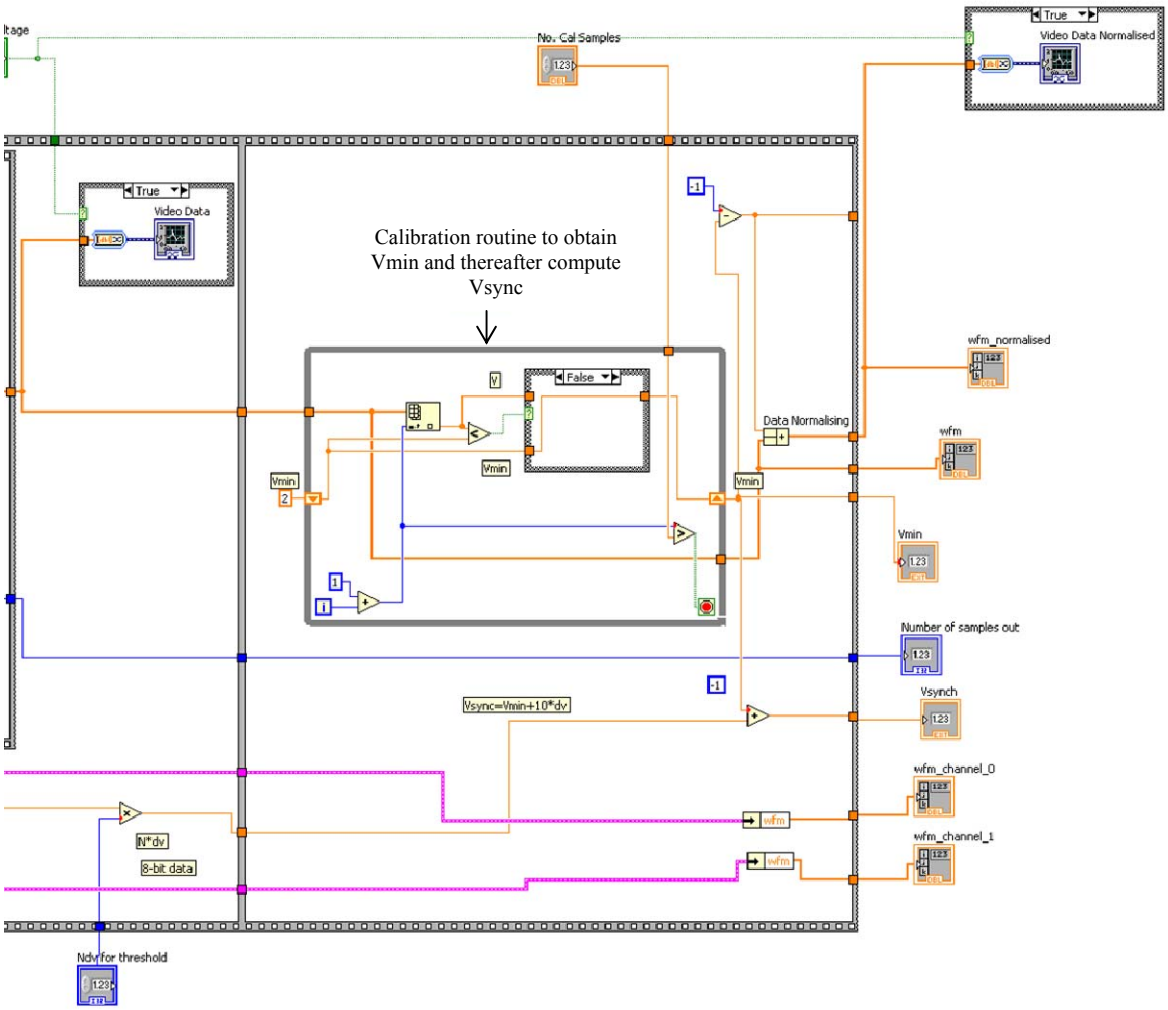



Figure 35. *Fetch From 5112* sub VI – Calibration routine to compute Vsync

### *c. NTSC Processor Sub VI*

The *NTSC Processor* sub VI is the crux of the NTSC decoding module, as it contains the logic involved in retrieving the NTSC formatted video from the incoming signal received by the array. Figure 36 shows a NTSC formatted video signal. This figure will be used as a basis for the explanation of *the NTSC Processor* sub VI. The following significant times are indicated in the figure: Point (a)—start of sync pulse, Point (b)—end of sync pulse, Point (c)—start of video data and Point (d)—end of video data.

Figure 37 shows the software flow chart of the *NTSC Processor* sub VI. The data record length is set long enough to ensure that at least one complete frame of data is included. When each record of data sampled from the 5112 is sent to this sub VI, it first attempts to find a valid sync pulse. This is done by first detecting a data point that has a value below Vsync, preceded by a data value greater than Vsync. This corresponds to point (a) in Figure 36. When this is found, the program will then look for data with the reverse condition where the value is greater than Vsync in the attempt to find point (b) of the same figure. When this data is found, the length of time in which the data points were less than Vsync is tested to ensure it is a valid pulse (*time* > *Tmin*, where *Tmin* is derived as the minimum time for it to be considered a sync pulse) and not a random value due to noise. Figure 38 shows this implemented in the *NTSC Processor* sub VI.

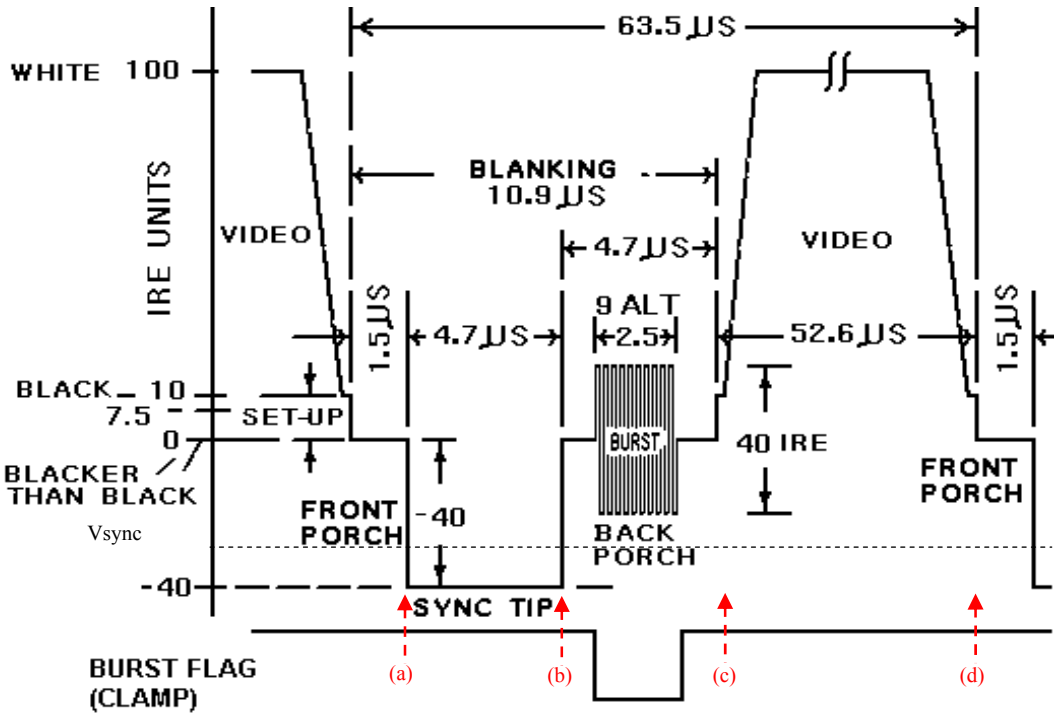



Figure 36. Detailed video timing

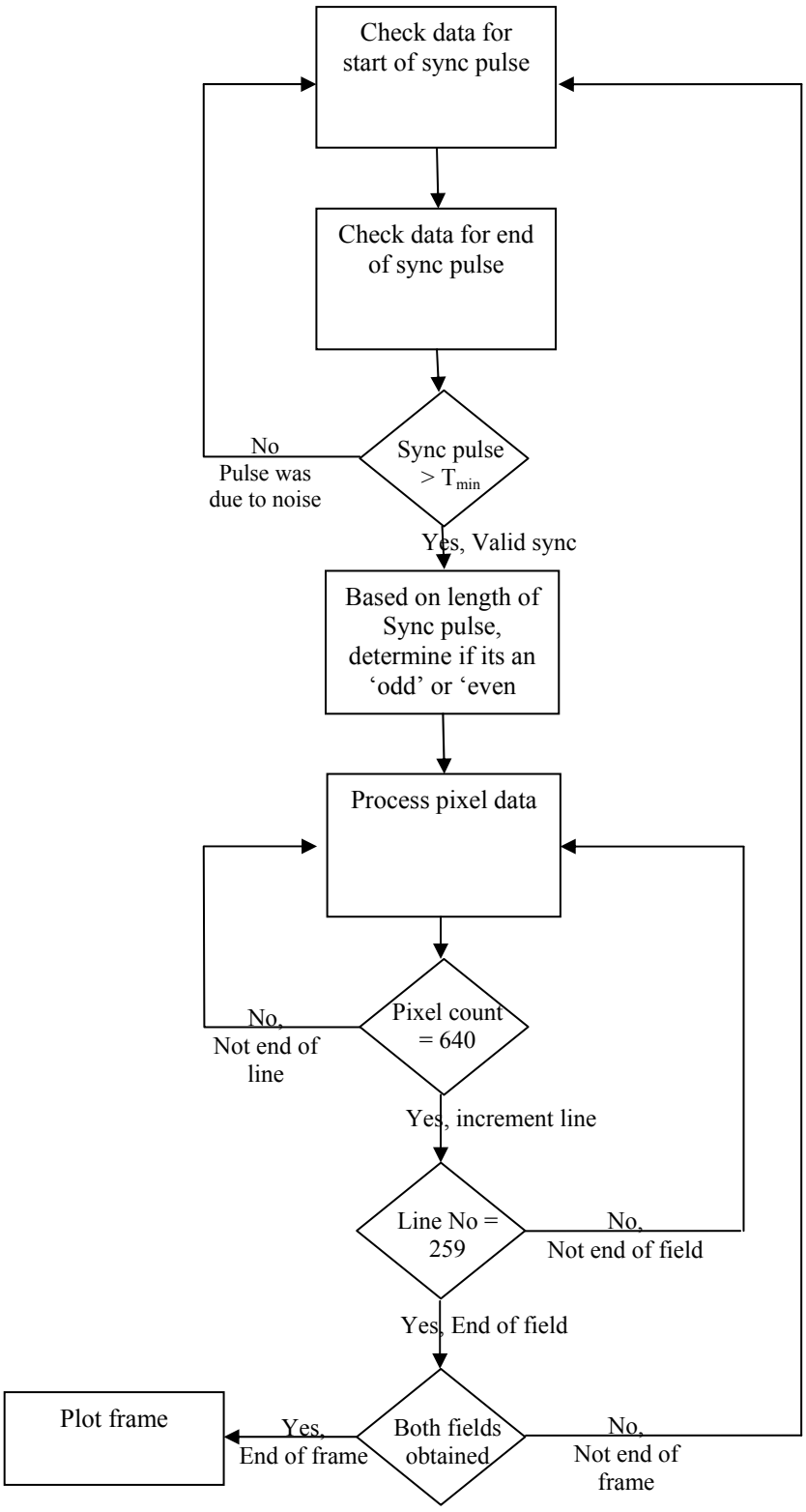



Figure 37. Software flow chart of *NTSC Processor* sub VI

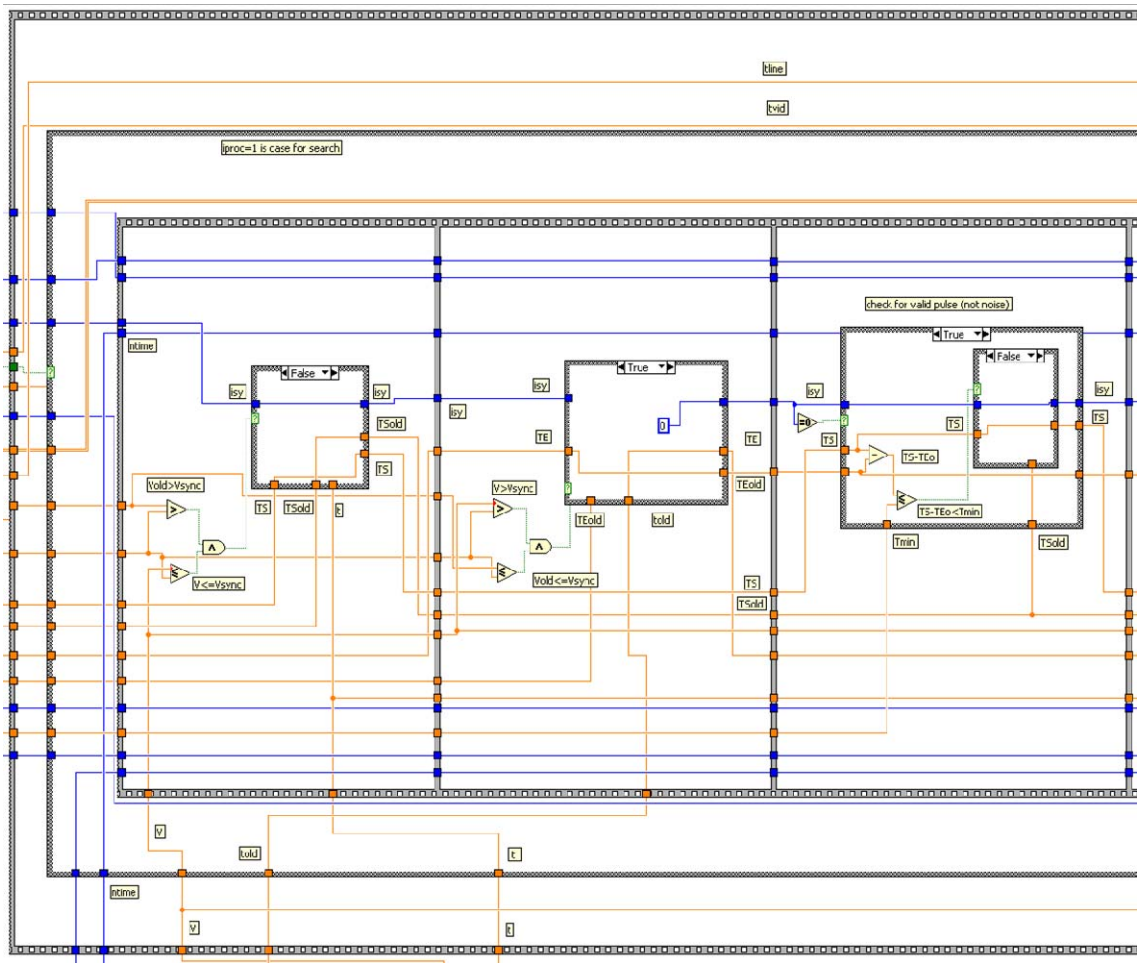



Figure 38. Determining a valid sync pulse in *NTSC Processor* sub VI

Once time (b) has been found, the length of time in which the data points were less than Vsync will be used to determine if the current field detected is an odd or even field. Using point (b) as a reference point, the program now knows the location of actual video information stored in the current data record that will need to be decoded. Figure 39 shows the routine that determines the type of field and initializes the variable for subsequent decoding of video data.

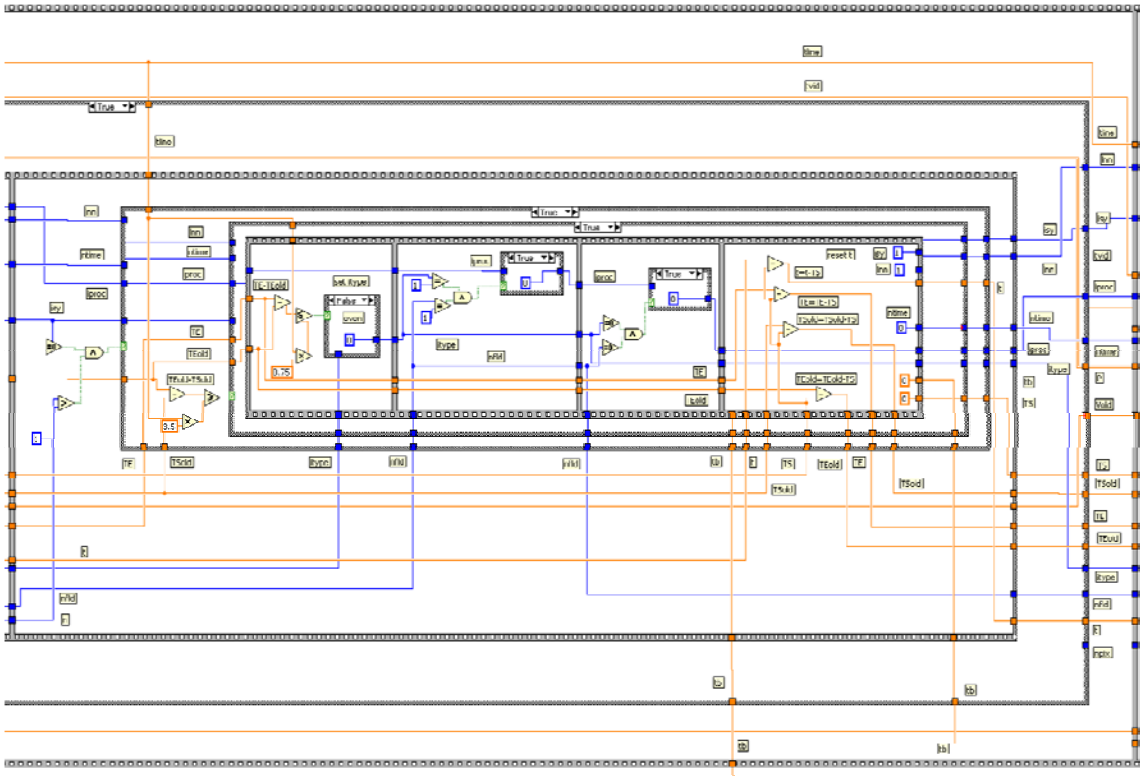


Figure 39. Determining the field type and initializing of variable for video data

From Figure 36, it can be seen that, from time (b) to time (c), there is a time delay of 4.7 µsec before the actual video data information is transmitted. This sets the start point of the video data stream. The software next reads the subsequent data stream, 640 pixels to the end of the line, 259 lines to the end of the field. When both fields of a frame have been obtained, the video image will then be plotted on the display screen. Figure 40 shows this final routine in the software.

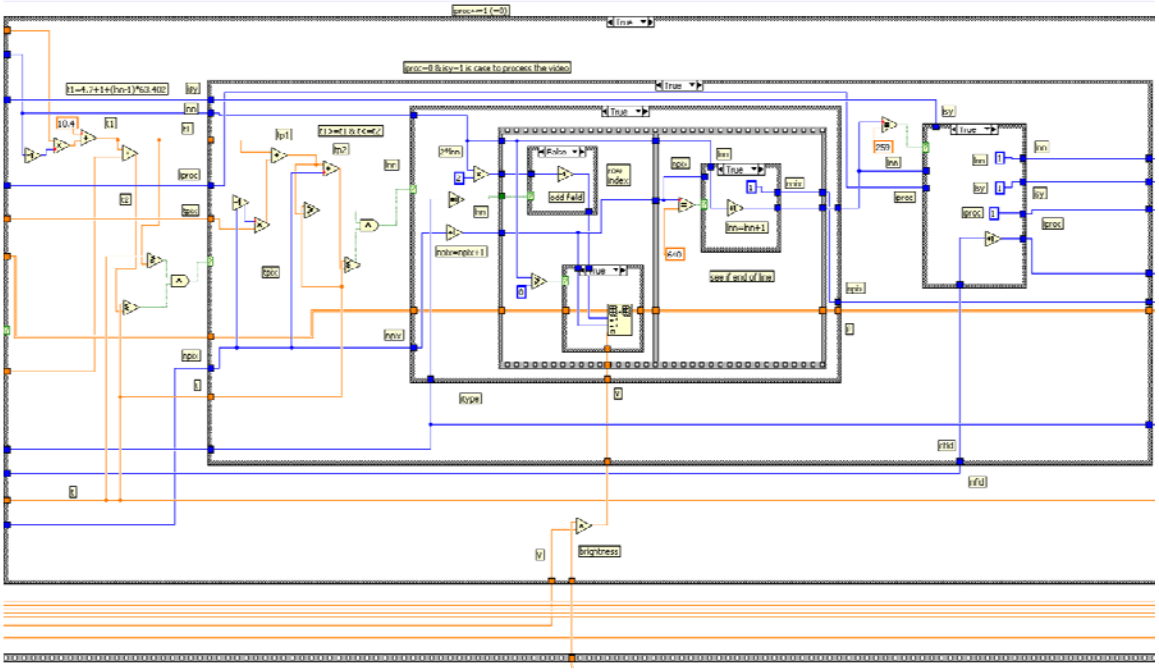


Figure 40. Obtaining the video information from the data stream

### 2. FM Demodulation Module

This module was developed based on the library function *FM demodulation* VI provided in the Signal Processing toolkit in LABVIEW. Using the *Open and Fetch From 5112* sub VI, a program was written for FM demodulation of data on a single channel. Figure 41 shows the front panel of the program.

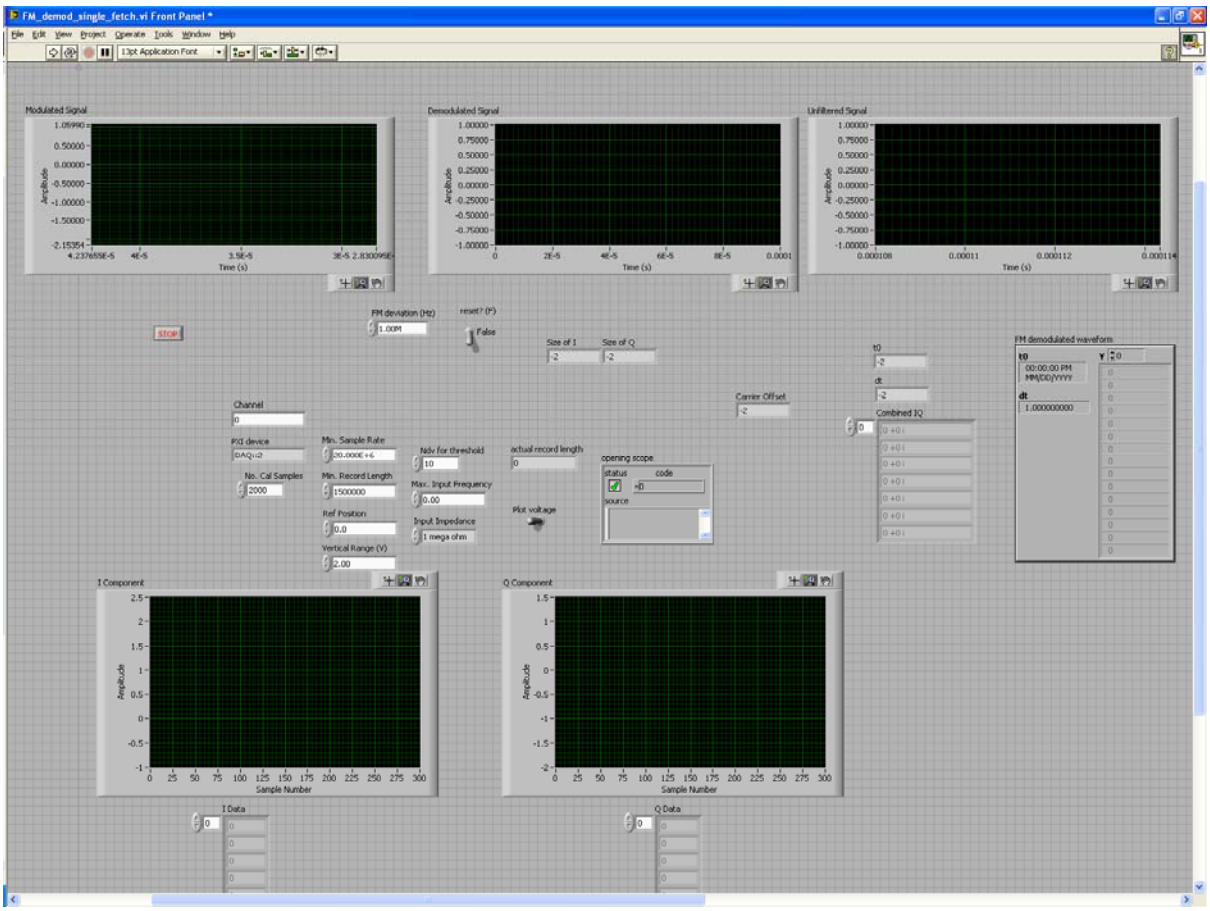



Figure 41. Front panel of FM demodulation software

The front panel displays the waveform of the incoming *I* and *Q* components, the modulated signal (complex envelope $I + jQ$), the demodulated signal before filtering and the demodulated signal after filtering. The settings required for *Open and Fetch From 5112* sub VI are similar to what was detailed in Section 1 on *NTSC Processor*. Figure 42 shows the FM demodulation module implemented for a single channel.

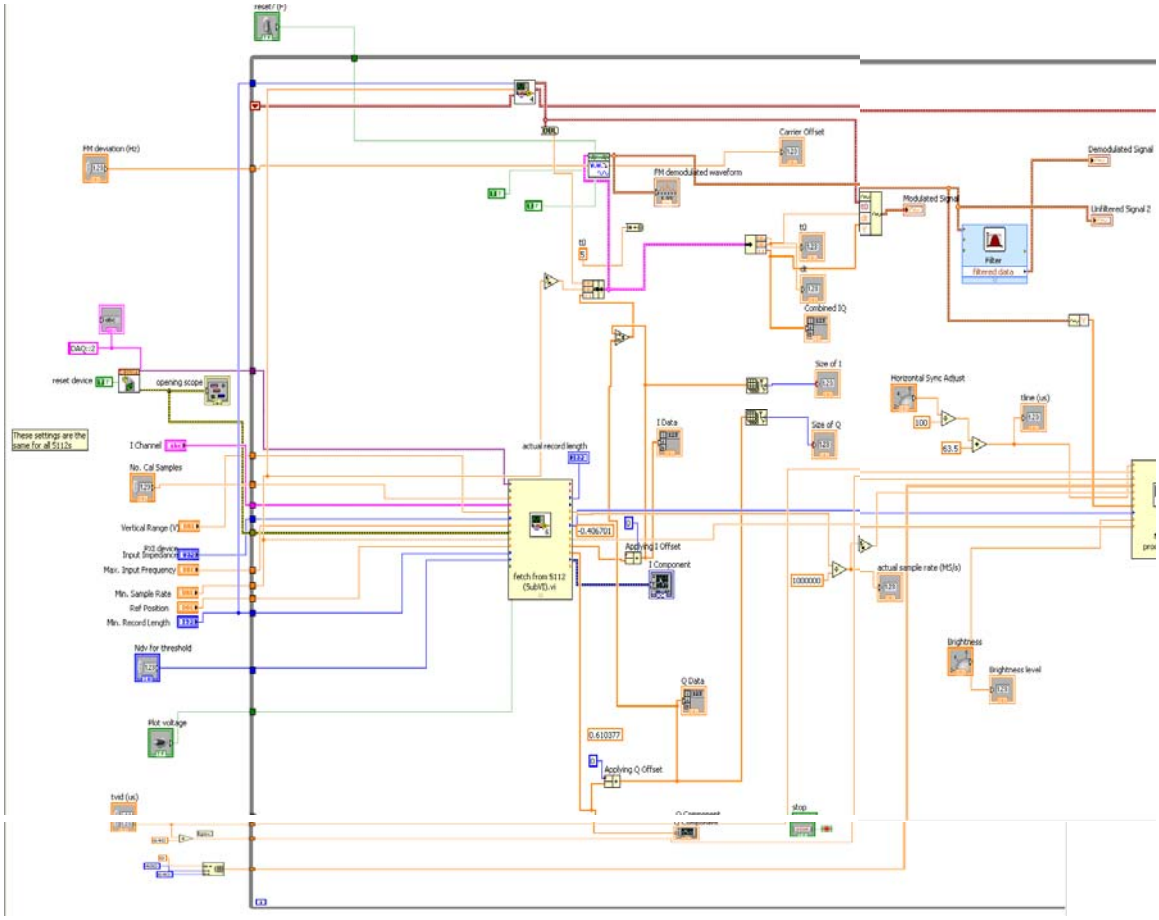



Figure 42. Detailed software implementation of FM demodulation

When the program is run, data from the 5112 channel is sampled and the clustered *I* and *Q* data is retrieved. This data is subsequently separated into the *I* and *Q* components before recombining them to obtain the complex representation of the incoming signal at baseband. This data is then sent to the *FM demodulation* VI. The output of the demodulation VI is subsequently filtered to retrieve the original message signal. The filter cutoff frequency programmed in the sub VI is dependent on the expected bandwidth of the incoming signal.

### 3. Tracking Module

In this section, the software implementation of the tracking module is covered. This module is an implementation of RSNS and monopulse DBF, the theory of which

was covered in detail in Chapter III. For the module developed, the algorithm has two means of tracking. For the first means of tracking, the module uses only RSNS to track the target. The RSNS system is based on two modului, [5, 9], with a dynamic range of approximately 36 bins. With a field of view of $180^\circ$, this will evaluate to $5^\circ$ per bin. This provides a good estimate of the direction of the target as it will provide an angular resolution of $5^\circ$ at broadside. In the second means of tracking, the system uses RSNS to obtain an intial scan angle before using monopulse DBF for subsequent tracking of target. Figure 43 shows the flow chart of the tracking antenna module.

### *a. DBF Simulation Program in MATLAB*

A MATLAB program was written to obtain the plots of sum and difference beams expected for an array of six elements. The codes are found in Section A of the Appendix. In [8], the design considerations for tracking of a UAV were presented and analyzed. It was concluded that for the tracking of a UAV, the maximum azimuth angle is approximately $40^\circ$ with the antenna tracking the UAV approximately on the horizon. At this scan angle, to prevent grating lobes, the spacing between the elements should not exceed 0.073 m. To remain consistent with the findings in [8], the element spacing was set at $\lambda/2$ apart. At 2.4 GHz, the spacing between elements is 0.065 m. Figure 44 and Figure 45 show the output of the sum and difference beams for scan angles of $\theta_s = 0^\circ$ and $\theta_s = 40^\circ$, respectively, for uniform amplitude weights. In both the plots, it can be seen at the designated scan angle, the pattern of the sum beam has a maximum while the difference beam has a value of zero. The linear region of the difference beam near the scan angle is used to compute the correction angle based on Equation (3.1). It is observed from the figures that the linear region has a span of approximately $20^\circ$ ($\pm 10^\circ$ from the scan angle). Thus, for monopulse DBF to be accurately used, the initial scan angle must not be greater than $20^\circ$ off from the true direction of the signal. This assures that the tracker is using the main beam null rather than a null outside of the main lobes. Since RSNS provides an angular accuracy of $5^\circ$, it provides a good estimate to obtain the initial scan angle (i.e., acquisition) before DBF is used for fine tracking.

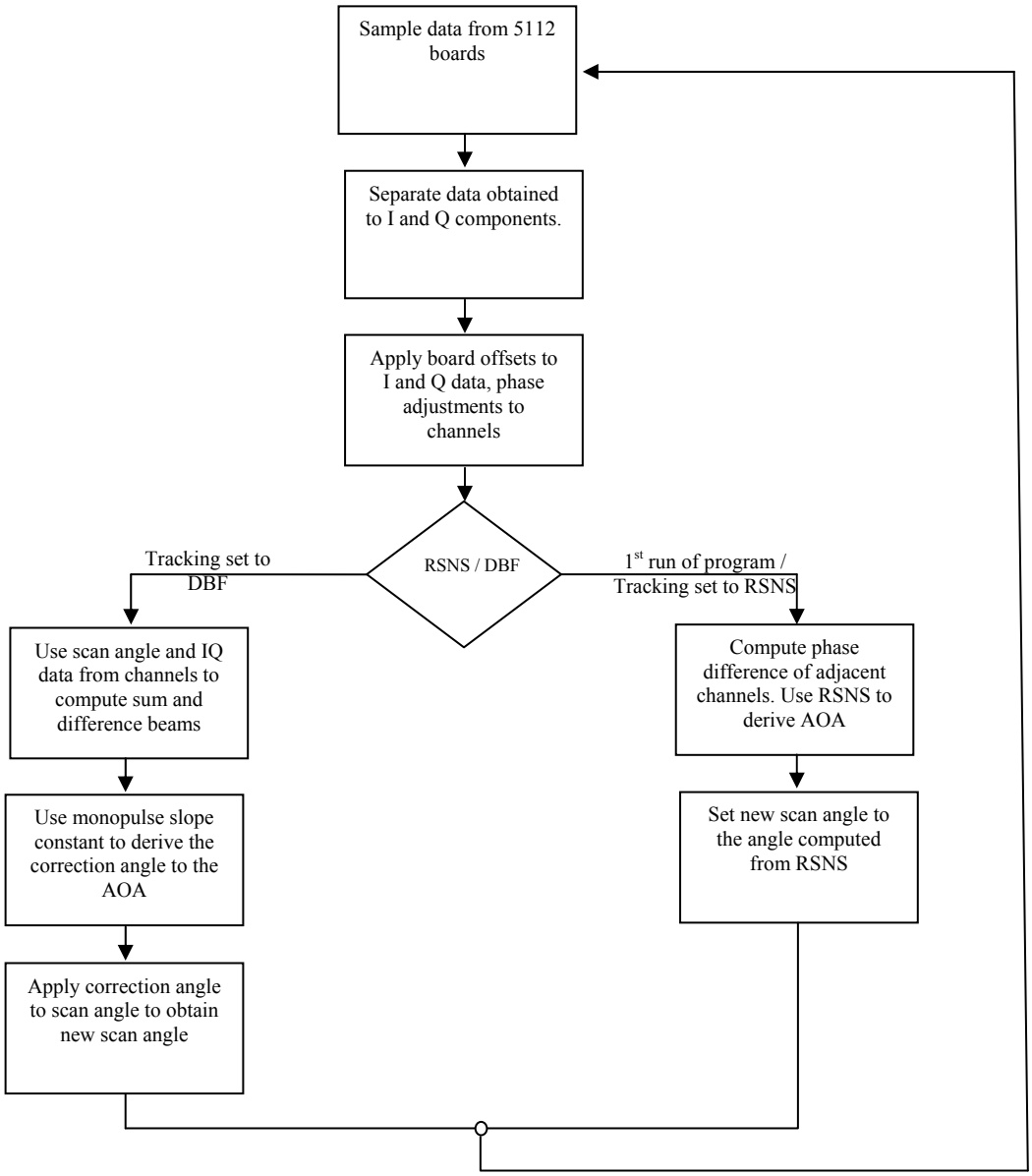



Figure 43. Flow chart of algorithm used in the tracking module

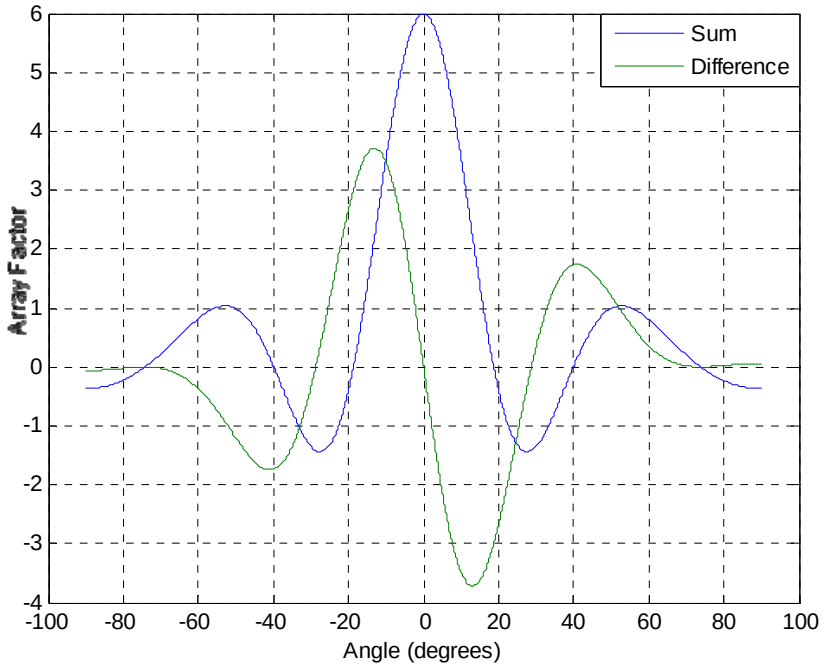



Figure 44. Plot of sum and difference beam at broadside ($\theta_s = 0^\circ$)

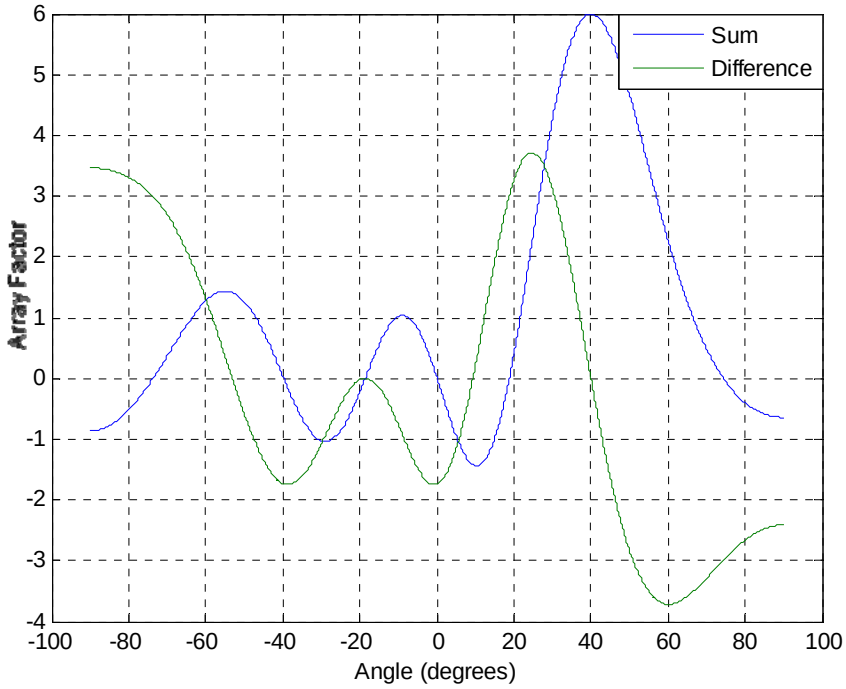



Figure 45. Plot of sum and difference beams at $\theta_s = 40^\circ$

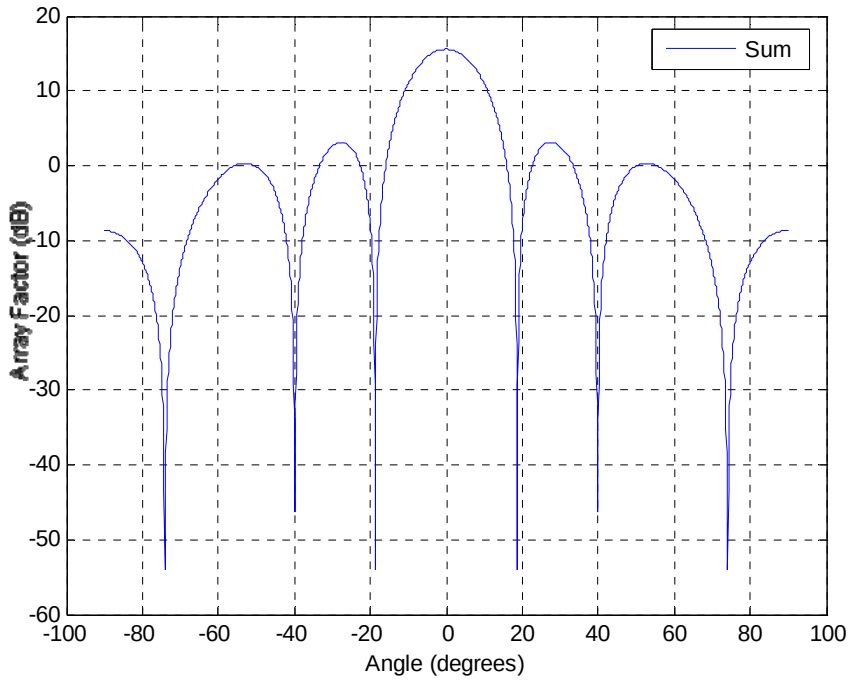



Figure 46. Plot of antenna pattern for sum bean at $\theta_s = 0^\circ$

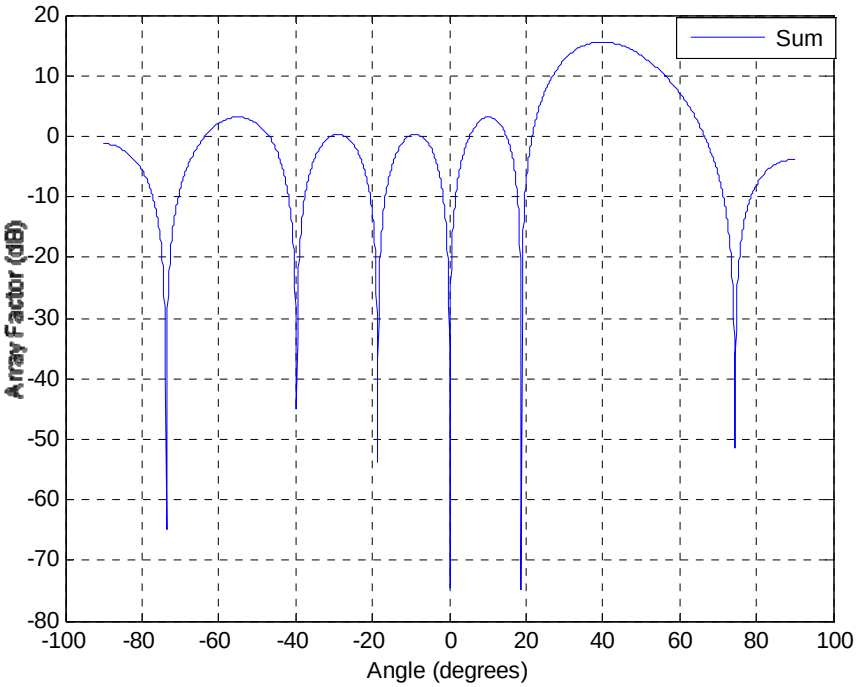



Figure 47. Plot of antenna pattern for sum bean at $\theta_s = 40^\circ$

The gain of the six-element antenna computed using Equation (3.18) was found to be approximately 15.5 dB. The actual implementation of the antenna array is expected to have a gain lower than this value due to losses not accounted for in this simulation.

Figure 48 and Figure 49 show the plots of the normalized $\frac{\Delta}{\Sigma}$ output against $\theta$. An important aspect in determining the correction angle is the value of the monopulse slope constant, *K*, which is the gradient of the linear region of the normalized difference/sum beam near the scan angle. The value of *K* changes with the scan angle. For Figure 48, at $\theta = 0^{\circ}$, the slope of the graph is -0.076 while the equivalent plot for $\theta = 40^{\circ}$ has a slope of -0.0578. If a different value of *K* is used other than the required value for a specific scan angle, the DBF algorithm will take a longer time to converge to the angle of arrival. To prevent overshoot and oscillations, the smaller value of *K* of -0.076 is used for this module.

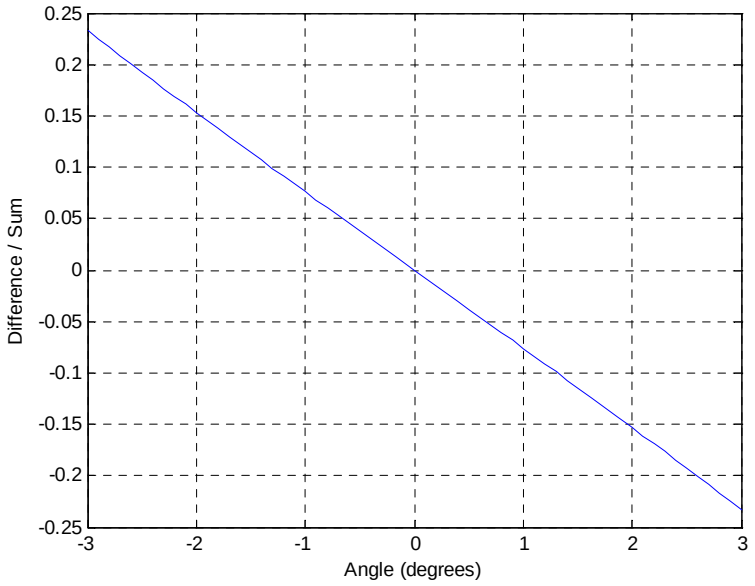



Figure 48. Plot of $\frac{\Delta}{\Sigma}$ when $\theta_s = 0^{\circ}$

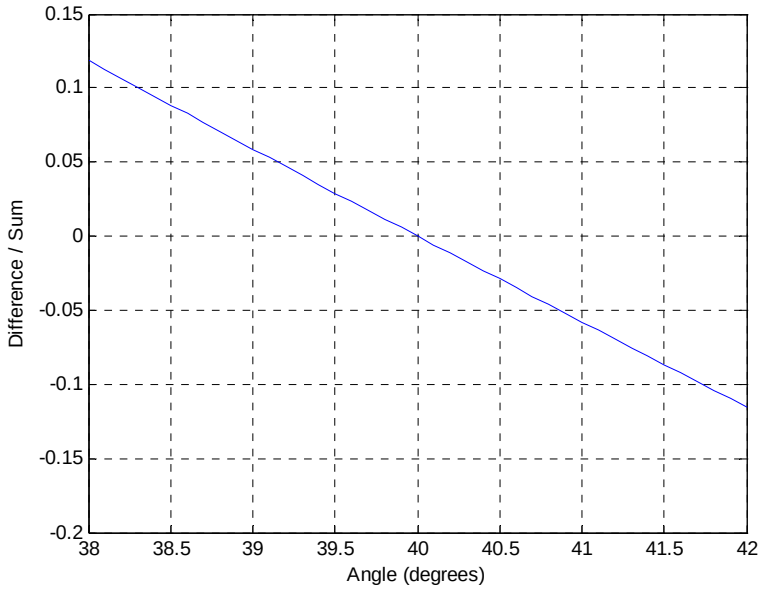



Figure 49. Plot of $\frac{\Delta}{\Sigma}$ when $\theta_s = 40^\circ$

### *b. LABVIEW Implementation*

Figure 50 shows the front panel of the tracking module developed for the system.

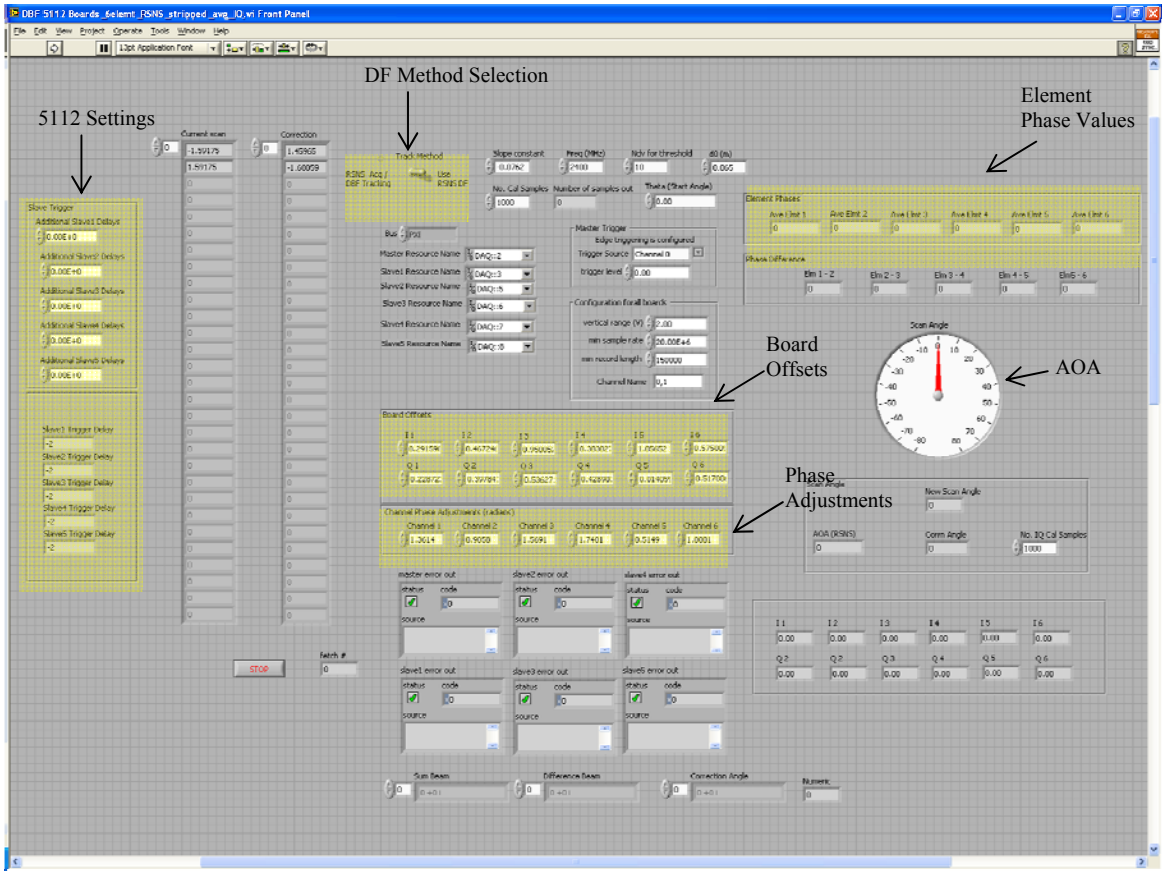



Figure 50. Front panel of tracking module

The first feature of the software is the control of the six PXI-5112 cards to sample the data from the output of the demodulator boards. Figure 51 shows the setup of this process in software. Channel[3] 1 is used as the master resource while the other channels are used as slave resources. The selection of the master resource is done arbitrarily, primarily to dictate the master reference signal so that the other channels are able to then use it as a reference. This ensures that the data sampled from each channel is correlated. For the master channel, an immediate reference trigger is used for the expected continuous data input. The triggering information for the slave channels is

[3] Channel refers to the set of *I* and *Q* data obtained from a 5112 board, i.e., an array channel, not a specific channel on the 5112 board.

derived from this master reference trigger. The connections for all the slave channels are set up identically. The LABVIEW resource and corresponding array channel (element) number are shown in Table 5.

Table 5. Mapping of channel number to resource name

| Master / Slave | Channel/ Antenna Element | Resource Name |
|---|---|---|
| Master | 1 | DAQ 2 |
| Slave | 2 | DAQ 3 |
| Slave | 3 | DAQ 5 |
| Slave | 4 | DAQ 6 |
| Slave | 5 | DAQ 7 |
| Slave | 6 | DAQ 8 |

### *c. Board Offsets and Phase Adjustments*

Once the 5112 data has been captured by the program, the individual *I* and *Q* components are separated. The board dc offsets, which have been computed using the AD8347 calibration software, are then added to the individual *I* and *Q* data for the different channels (Figure 52). The calibration data is used to remove any inherent dc offset that is introduced by the demodulator boards.

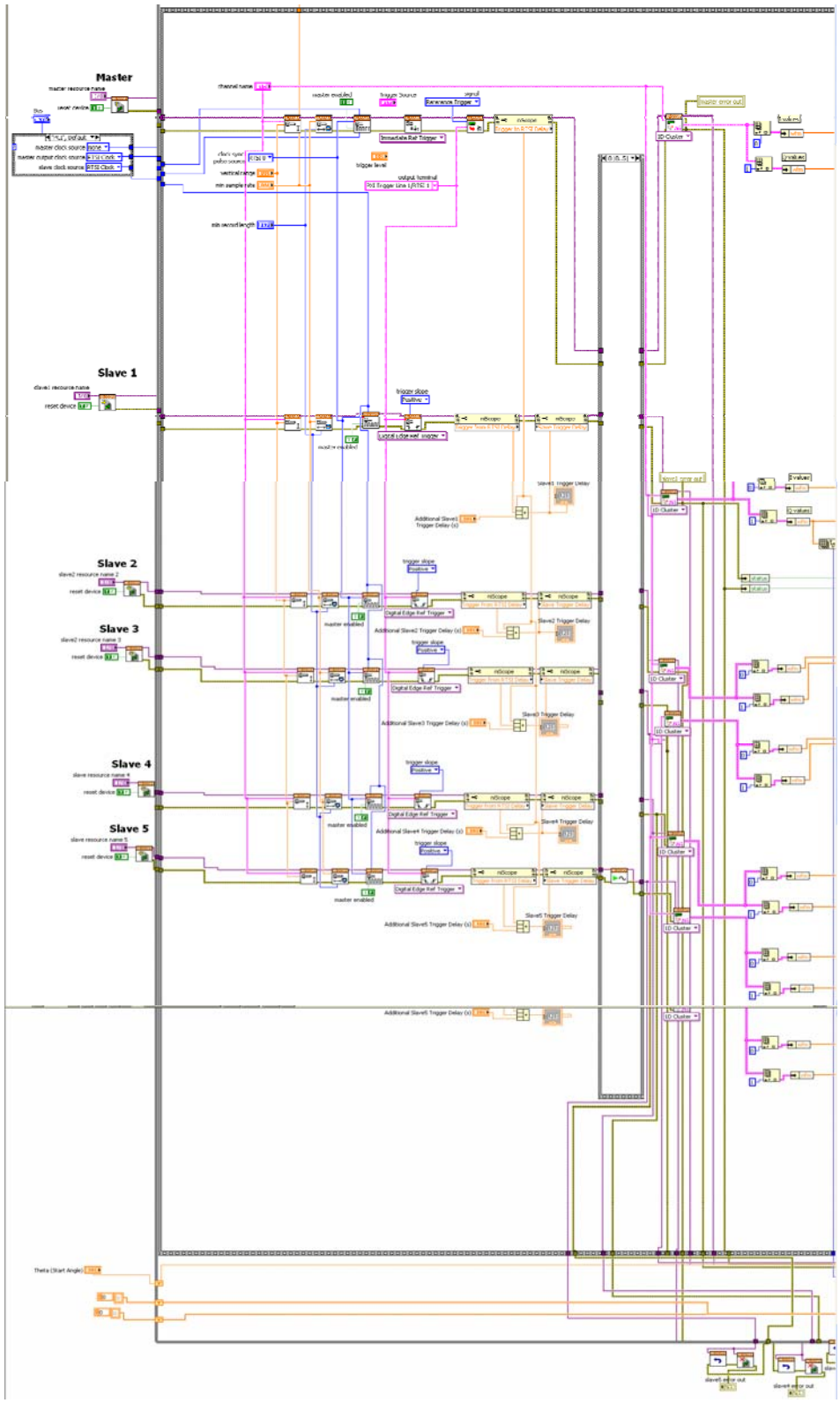



Figure 51. Data acquisition in LABVIEW

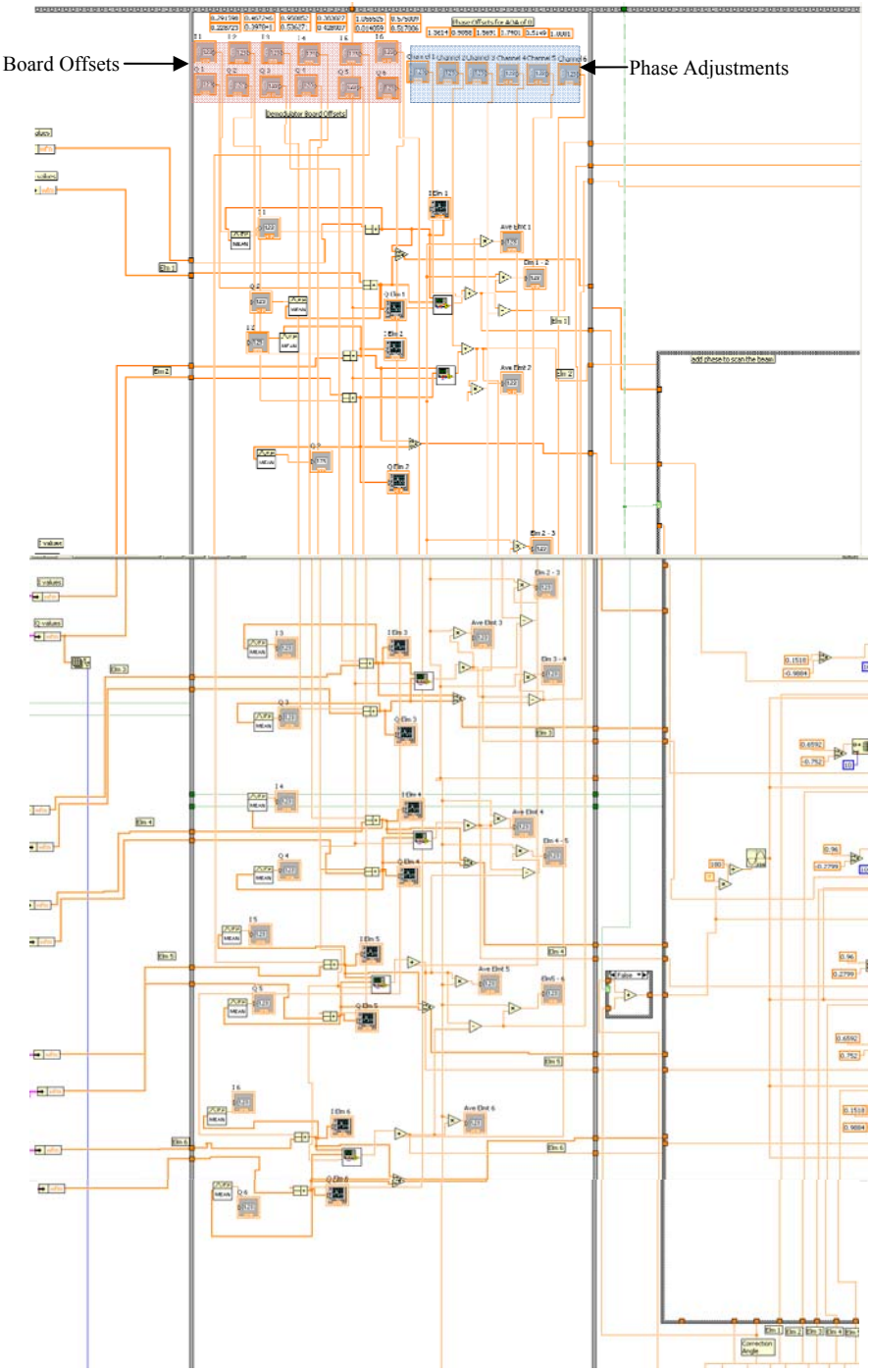



Figure 52. Application of calibration data offsets and phase adjustments

Ideally, once the dc offsets have been accounted for, each channel should have similar phase values for a plane wave that is incident normal to the array setup. This

was simulated in a lab test setup using a 1:6 power divider so each array element was fed the same signal input to simulate a normal incident wave. The channels did not have an equal phase. This could be attributed to the different insertion losses on the connectors and RF wires used to connect the hardware elements to the 5112. To account for the phase differences, a calibration must be conducted prior to the use of the tracking module. A plane wave of normal incidence has to be transmitted to the array elements. The phase of each of the channels will then be recorded and entered in as phase adjustments. This ensures that each channel has a common phase reference for a wave received at normal incidence.

### *d. Scan Angle Computation*

The next block of code computes the scan angle. When RSNS is selected as a means for tracking, the element phases are first computed. The phase differences between adjacent elements 1–2, 3–4, and 5–6 are then found and used in the computation of the angle of arrival. For RSNS with virtual spacing [10], the RSNS requires only two elements to compute the direction of angle of arrival. For the implementation here, instead of using a single phase difference between two elements, the average of three phase differences (1–2, 3–4, and 5–6) is used to compute the angle of arrival as shown in Figure 53. This reduces direction finding error due to element phase errors.

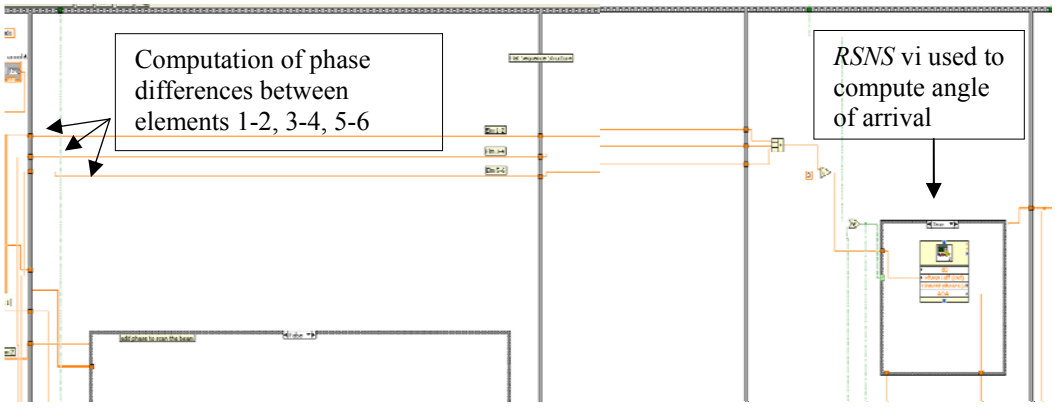



Figure 53. Computation of phase differences for RSNS

When monopulse DBF tracking is selected, the RSNS routine is executed for the first iteration of the loop in the program to determine the initial scan angle. As discussed previously, for DBF to be used for monopulse tracking, the deviation of the scan angle and the actual angle to the target has to be $\leq 20^{\circ}$. Once the initial scan angle is obtained through RSNS, the DBF equations listed in Chapter III, Section C are then used to compute the array factor for each element. Figure 54 shows the array factor computation for each element.

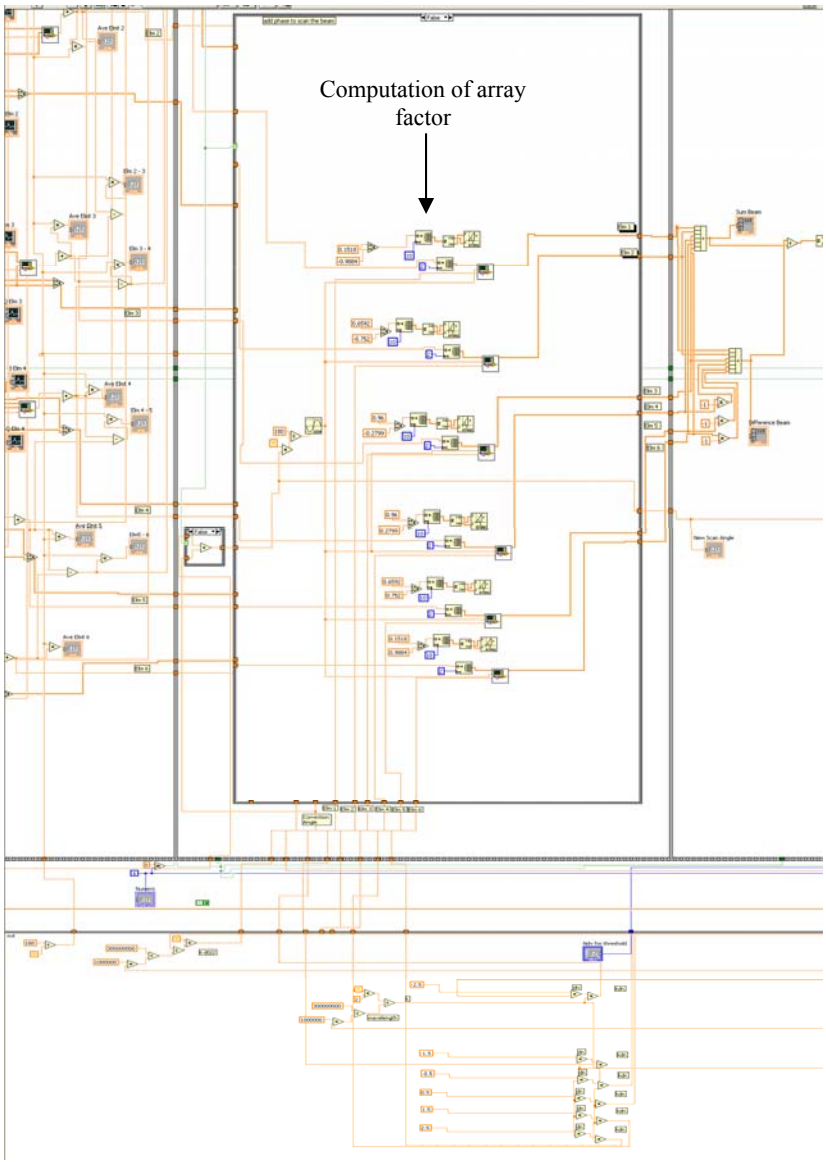



Figure 54. Computation of array factor for each element

Once the array factors for each element have been computed, the sum and difference of these factors are derived. Using these values and the monopulse slope constant, the correction angle for the direction of scan is then computed. This correction angle is subsequently added to the current scan angle to determine the new scan angle. In the next iteration of the loop, this new scan angle is used for the computation of the array factor of the elements. Ideally, when the target is in the direction of the current scan angle, the correction angle will approach zero.

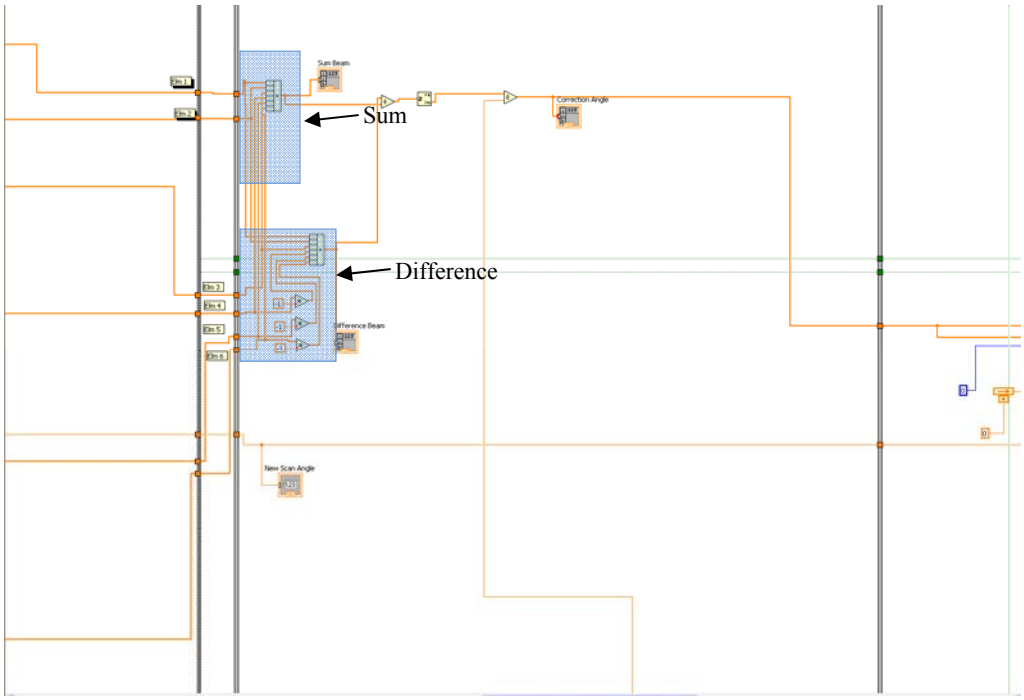



Figure 55. Computation of sum and difference beam to determine correction angle

## D. SYSTEM PERFORMANCE ANALYSIS

Based on the specifications of the building blocks of the receiver, the performance of the system is analyzed. A block diagram of the RF portion of the receiver is shown in Figure 56.

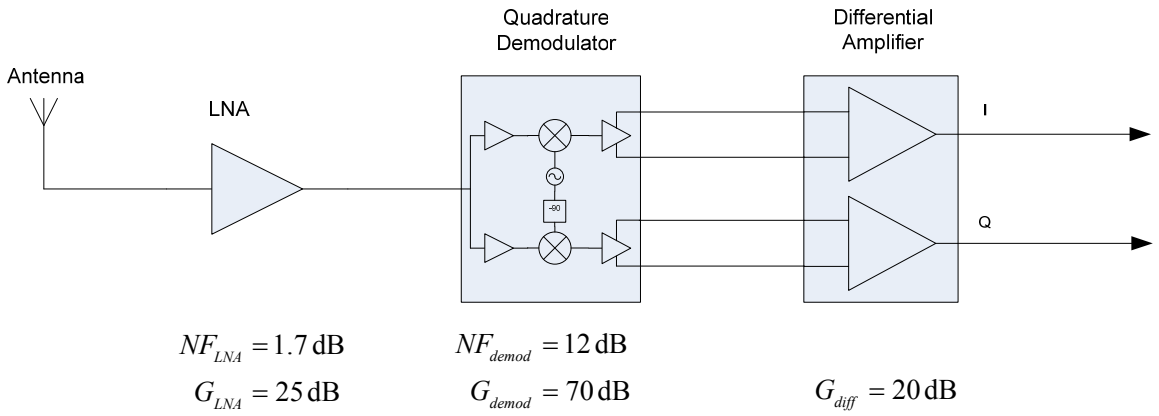



Figure 56. Block diagram of RF portion of the receiver

The equivalent temperatures of the LNA $\left(T_{LNA}\right)$, AD8347 demodulator $\left(T_{demod}\right)$, antenna $\left(T_A\right)$, and receiver system, $\left(T_{sys}\right)$, can be calculated using the following equations:

$$T_{LNA} = 290\left(F_{LNA} - 1\right) = 148.9\,\text{K}$$

$$T_{demod} = 290\left(F_{demod} - 1\right) = 4099.3\,\text{K} \tag{5.10}$$

$$T_A = \frac{hf}{k_B}\left(\frac{1}{e^{\frac{hf}{k_B T}} - 1} + 1\right) + \frac{7\times 10^{26}}{f^3} + T + \frac{f^{2.5}}{10^{25}} = 293.4\,\text{K} \tag{5.11}$$

where $h$ is Planck's constant, $k_B$ is Boltzman's constant and $f$ is frequency. The antenna temperature $T_A$ is essentially the weighted average of the noise temperature of the various noise sources that an antenna is "looking at," with the antenna power gain as the weighting factor. The equivalent noise temperature of the entire receiver system is:

$$T_{sys} = T_A + T_{LNA} + \frac{T_{demod}}{G_{LNA}} = 445.3\,\text{K} \tag{5.12}$$

Subsequently, the minimum power per channel $\left(P_{\min_{ch}}\right)$ that the receiver can receive (for correct demodulation) is given by:

$$P_{\min_{ch}}\left(\text{dB}\right) = P_{\min_{demod}}\left(\text{dB}\right) - G_{LNA}\left(\text{dB}\right) = -90\left(\text{dBm}\right) = -120\left(\text{dB}\right) \tag{5.13}$$

where $P_{\min_{demod}} = -65\,\text{dBm}$ is identified as the start of linear region of TPC30 on the AD8347 datasheet.

Maximum range is determined by $P_{\min_{ch}}$, below which the demodulation process cannot take place. Given typical transmit parameters on the UAV $\left(P_t = 0.1\,\text{W},\ G_t = 3\,\text{dB}\right)$ and the array element gain $G_r = 10\,\text{dB}$, the maximum range with the LNA is calculated to be:

$$R = \left(\frac{P_t G_t G_r \lambda^2}{\left(4\pi\right)^2 P_{\min_{ch}}}\right)^{1/2} = 34.4\,\text{km} \tag{5.14}$$

As the relationship between signal power and range is non-linear (signal power falls at a rate of $1/R^2$), Equation (5.14) also clearly shows that every doubling of range requires four times (increase of 6 dB) more power. The maximum range without the LNA is calculated to be 1705.8 m. Even though the receiver will work without the LNA, it can be concluded that the LNA is an essential part of the receiver to achieve practical maximum ranges.

Minimum signal-to-noise (SNR) ratio occurs when the signal is at its weakest, i.e., at $P_{\min_{ch}}$. Hence, the SNR can be calculated by:

$$SNR_{ch} = \frac{P_{\min_{ch}}}{N_{sys}} = 4.4\,\text{dB} \tag{5.15}$$

where, $N_{sys} = kT_{sys}B$. This calculation is for the SNR when there is only one antenna element. In the receiver system, there are a total of six antenna elements. Therefore, after coherent beamforming, the SNR will improve by a factor of *N*, where *N* is the number of antenna elements. Therefore,

$$SNR_{sys} = SNR_{ch} \times N = 12.2(\text{dB}) \tag{5.16}$$

In order to maximize the elevation coverage of the antenna, the antenna is pointed up about $15^\circ$ as illustrated in Figure 57. Hence, if the boresight of the antenna is referenced to $0^\circ$ (the antenna normal), the half-power beamwidth spans from $-15^\circ$ to $+15^\circ$.

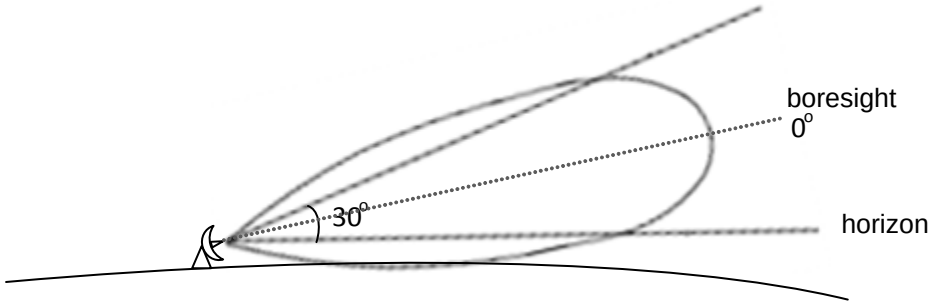



Figure 57. Diagram to show antenna tilt to maximize elevation coverage

A calculation was also done to determine the range of the system at this antenna elevation tilt for various transmit powers from 100 mW to 2 W, and at various elevation angles from $-15^\circ$ to $+75^\circ$. The results for the case without LNA, are plotted in Figure 58 while the case with LNA is plotted in Figure 59. Figure 59 shows an increase in range to roughly 35 km for the same transmit power of 100 mW when an LNA is used.

Figure 60 shows the level of received power versus the distance of the UAV if the receiver array does not make use of an LNA at the front end. A dramatic increase in power is obtained with the addition of an LNA as shown in Figure 61.

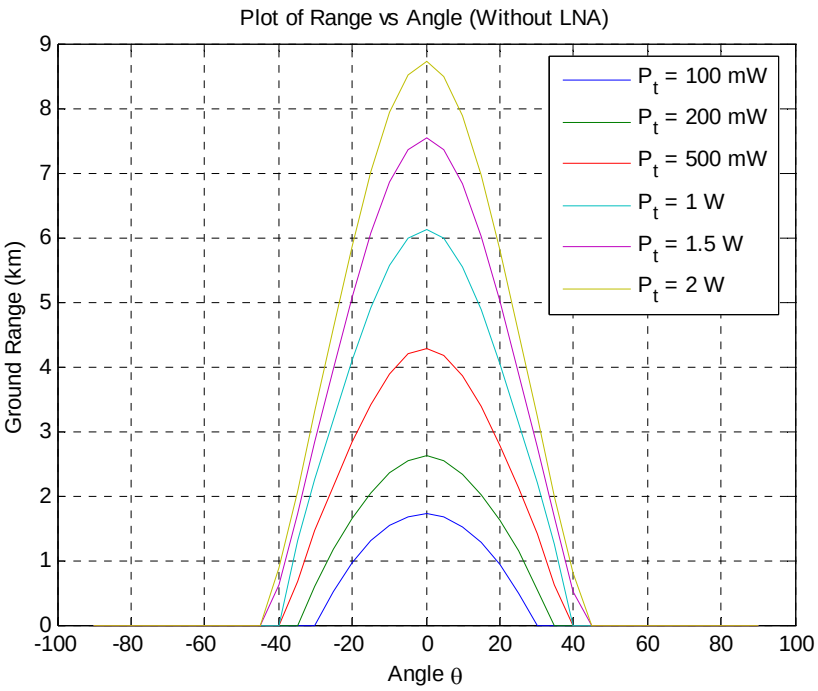



Figure 58. Plot of maximum range versus elevation angle (without LNA)

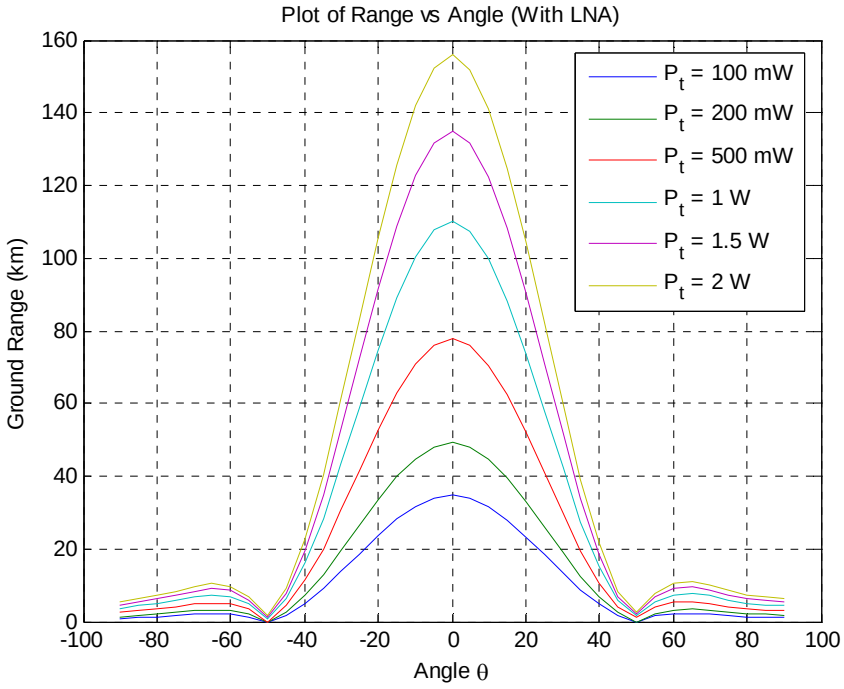



Figure 59. Plot of range versus elevation angle (with LNA)

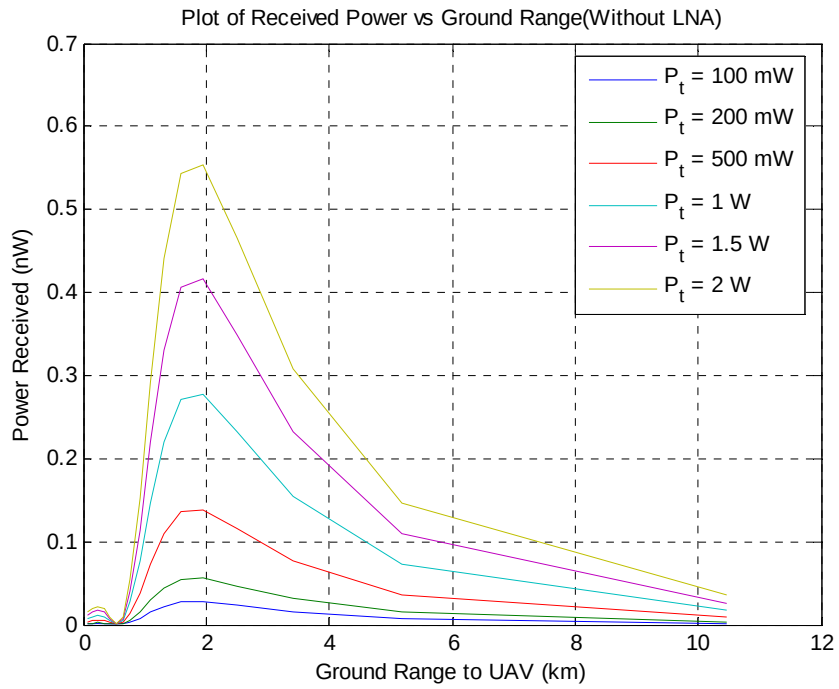



Figure 60. Plot of received power against ground range (without LNA)

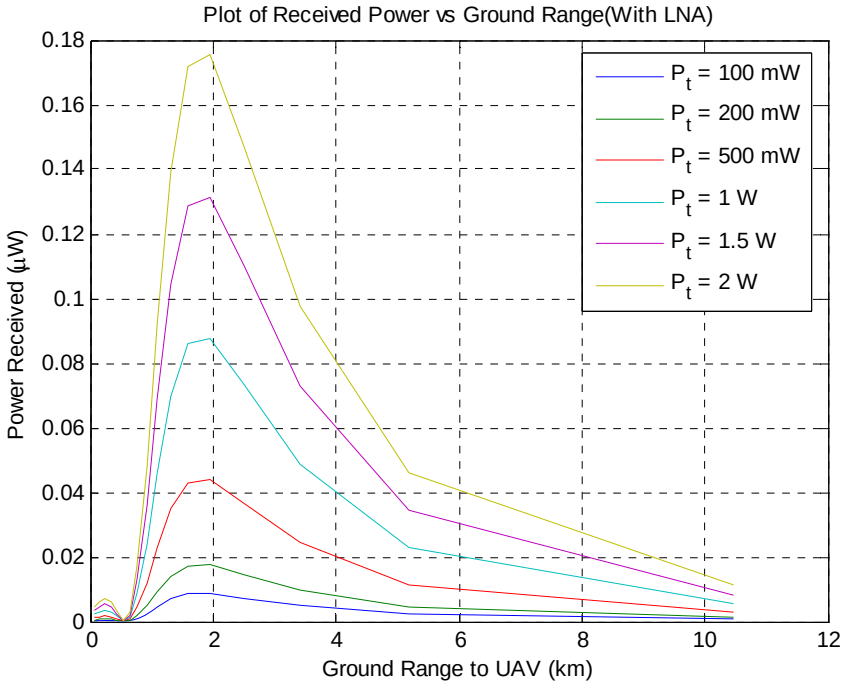



Figure 61. Plot of received power against ground range (with LNA)

Using the $4/3$ earth model, the radar horizon, when the UAV is flying at $H_t = 3000\,\text{ft}$, is calculated to be $R_{RH} = \sqrt{2R_eH_t} = 125\,\text{km}$. $R_e$ is the equivalent earth radius and can be calculated by multiplying the actual earth's radius by $4/3$. A Matlab program written to calculate and plot the various performance parameters can be found in Section C of the Appendix.

This chapter presented the hardware and software components of the system in detail and provided an analysis of the system performance. The next chapter discusses the testing and validation of each of the hardware and software components developed for the digital tracking array.

THIS PAGE INTENTIONALLY LEFT BLANK

# VI. DESIGN VERIFICATION AND VALIDATION

This chapter discusses the testing and validation of each of the hardware and software components developed for the digital tracking array. Section A gives an overview of the results obtained from the hardware testing of the two-element subarray antenna and presents the antenna pattern of the array. Section B covers the testing of the three software modules developed for antenna processing. Section C covers the integration of the hardware and software components for the purpose of acquisition and tracking and details the test conducted for the array in the anechoic chamber. Section D gives a summary of the results obtained.

## A. HARDWARE TESTING AND RESULTS

### 1. Two Element Subarray Antenna Performance Measurement

#### *a. Gain Measurement*

Based on the reciprocity theorem, the measurement of an antenna pattern can be done in either transmit or receive case, since the consequence of reciprocity is that the antenna will have identical transmit and receive patterns.

There are also many different methods for measuring the gain of an antenna. In our case, we have used the simplest and most common method, which is called the gain comparison method [36]. The method consists of comparing the power received by a reference antenna ( $P_{Ref}$ ) to the power received by the two-element subarray ( $P_{Test}$ ) at 2.45 GHz. The reference antenna used in this case is the Narda 645, which is a standard gain horn. These standard gain horns are precision horns designed to be used as standards for calibrating other antennas [37]. The gain of the reference antenna is known (Figure 62) and thereafter, the gain of the two-element subarray ( $G_{Test}$ ) can then be easily calculated by the formula:

$$G_{Test} = \frac{P_{Test}}{P_{Test}} G_{Ref} \tag{6.1}$$

or, when the quantities are in dB:

$$G_{Test(dB)} = P_{Test(dB)} - P_{Ref(dB)} + G_{Ref(dB)} \tag{6.2}$$

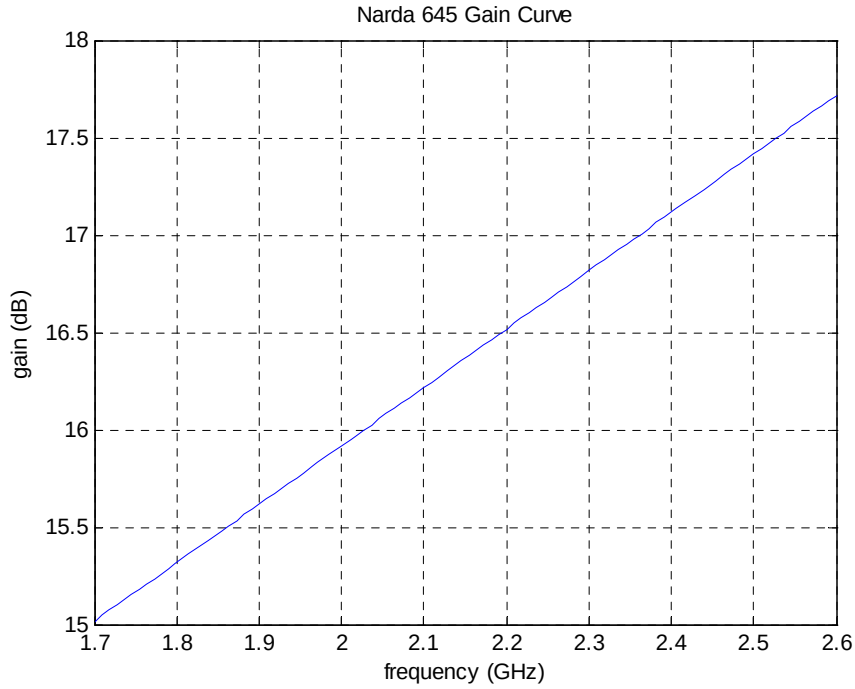



Figure 62. Gain of reference antenna (After [25])

### *b. Antenna Pattern Measurement*

Figure 63 illustrates the anechoic chamber measurement setup. The anechoic chamber is setup to measure $P_{Test} - P_{Ref}$ or the system loss (SL) of the antenna under test (AUT).

Once the antenna has been installed in the desired polarization (horizontal or vertical), the test program is started. During the test, the AUT is rotated on a pedestal through the desired angles. Figure 64 shows the reference antenna installed on the rotating pedestal. With the measurement for SL completed, Equation (6.2) can be used to calculate the gain of the AUT.

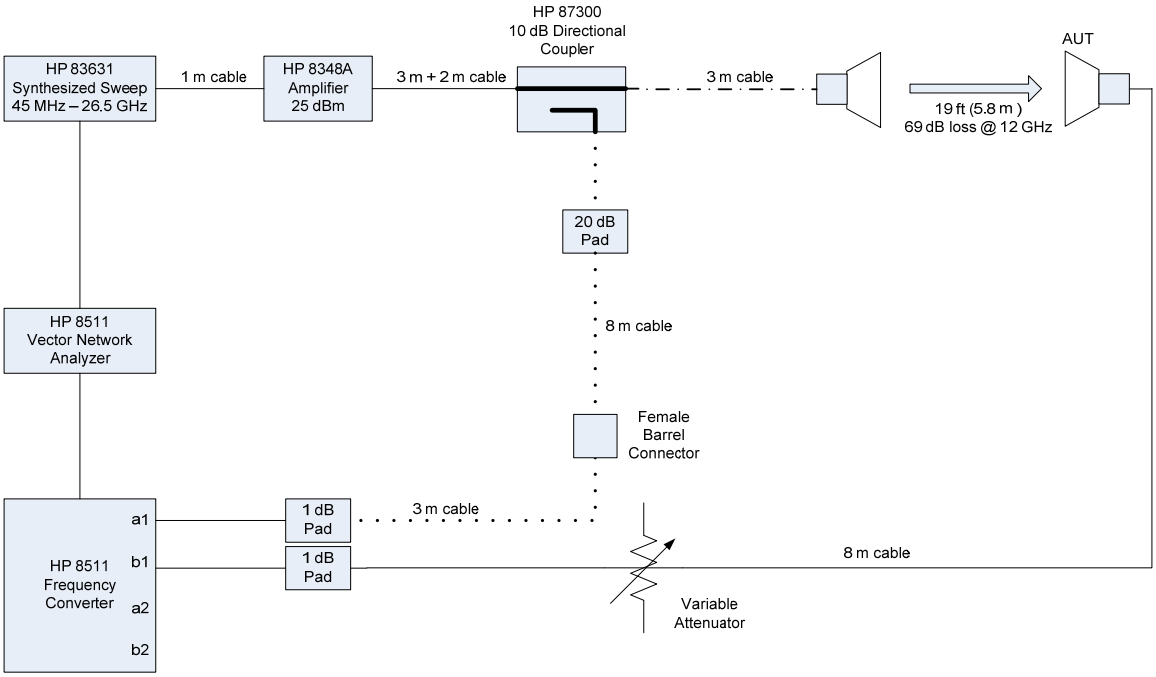



Figure 63. Measurement setup of the anechoic chamber

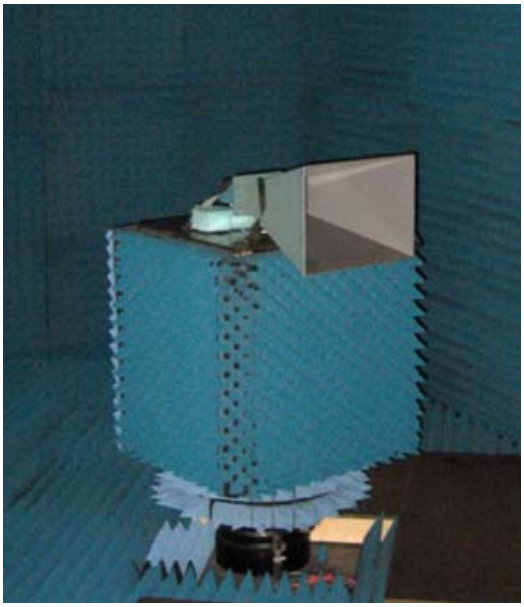


Figure 64. Reference antenna (Narda 645) setup to measure the horizontal plane on the rotating pedestal

The measured system loss of both the reference and two-element subarray mounted in the vertical and horizontal planes, as shown in Figure 65, were saved in a data array format. Matlab was then used to calculate and plot the gain of the two-element subarray. The Matlab code used in the calculation of the gain can be found in Section B in the Appendix. The gain of the two-element subarray is estimated to be around 10 dB. This is close to the calculated theoretical values using:

$$\begin{aligned} G_{sub-array} &= G_{dipole} \times 2 \times 2 \\ &= 6.56 \\ &= 8.2\,\text{dB} \end{aligned} \tag{6.3}$$

where the gain of a dipole, $G_{dipole} = 1.64 = 2.15\,\text{dB}$. The factors of 2 are to account for the two dipoles and also the presence of the ground plane. The principal plane cuts are plotted in Figure 67 and Figure 68. It is worth noting that the patterns do not look symmetrical due to the non-symmetric layout of the anechoic chamber. Figure 66 illustrates the layout of the NPS anechoic chamber. Note the asymmetry of the side toward the wedge-shaped region of the chamber.

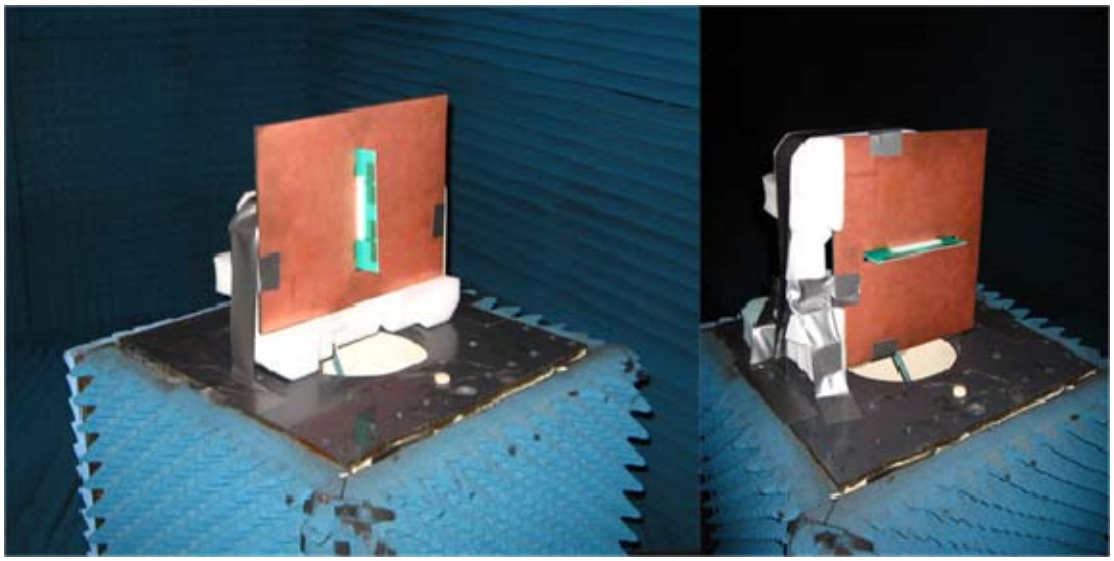


Figure 65. Two element subarray oriented in the in vertical (left) and horizontal (right) planes, which are used to obtain azimuth and elevation patterns, respectively

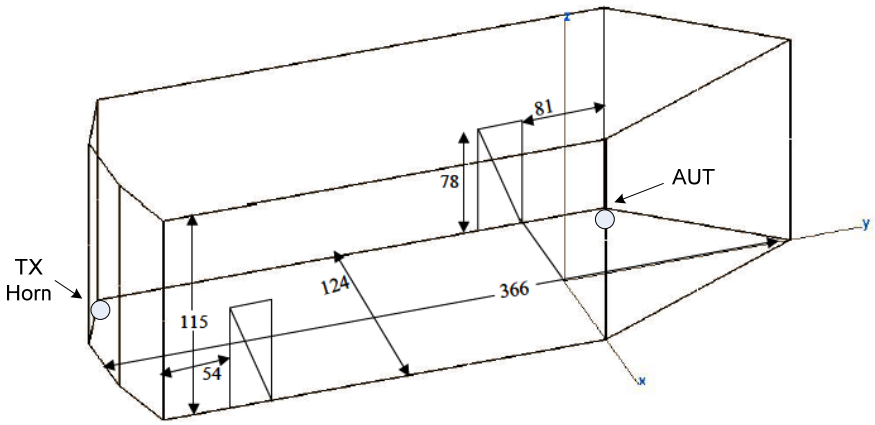



Figure 66. Layout of the NPS anechoic chamber (From [38])

Nonetheless, the measured results in Figure 67 and Figure 68 compare favorably with the values simulated via CST Microwave Studio and shown in Figure 69 and Figure 70, respectively. Azimuth is the horizontal plane; elevation is the vertical plane.

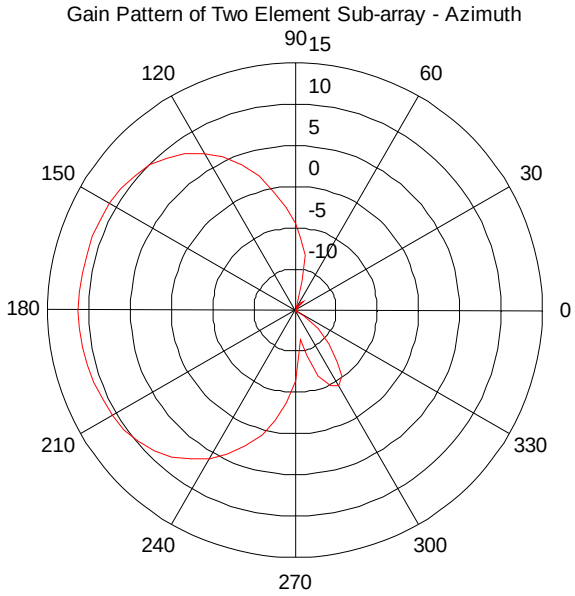



Figure 67. Measured azimuth gain pattern

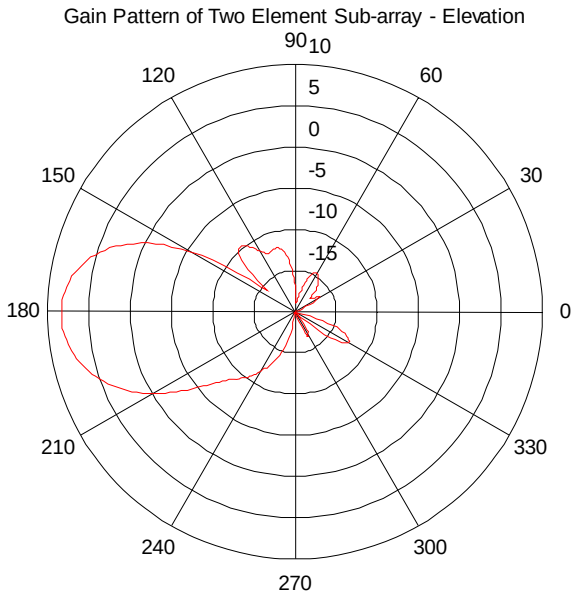



Figure 68. Measured elevation gain pattern

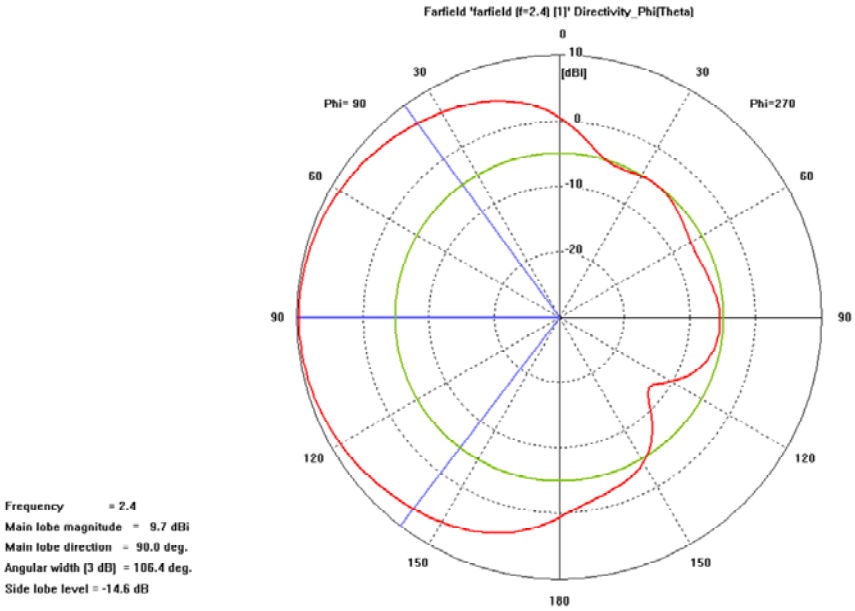



Figure 69. Simulated azimuth gain pattern using CST Microwave Studio

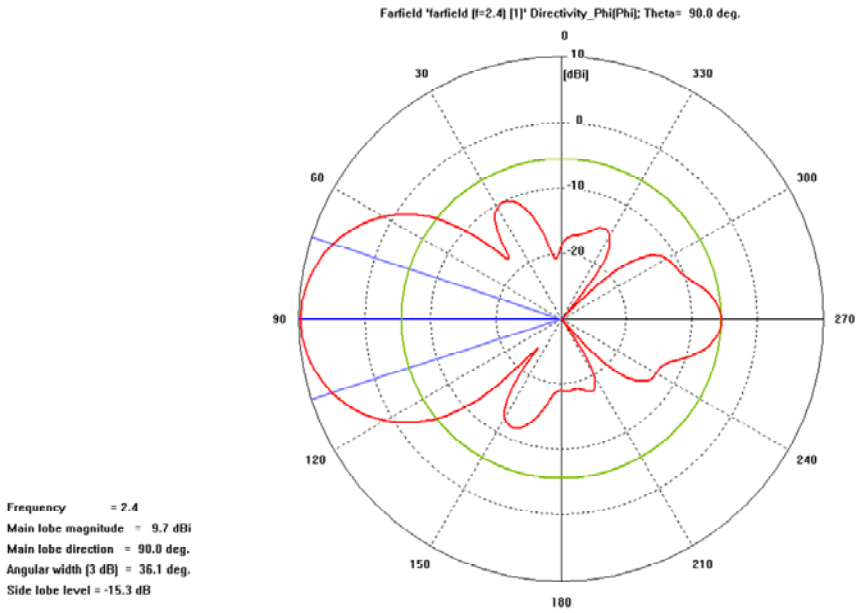



Figure 70. Simulated elevation gain pattern using CST Microwave Studio

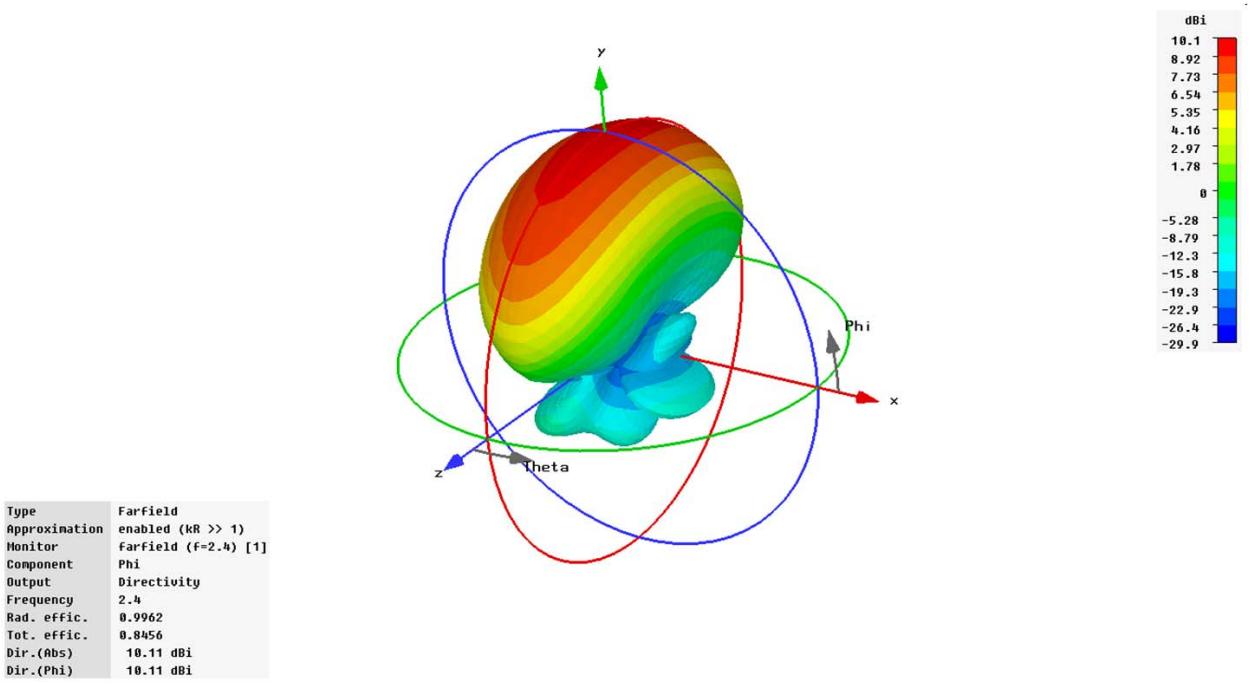



Figure 71. Simulated 3D gain pattern using CST Microwave Studio

### *c. Return Loss Measurement*

In addition to measuring the radiation pattern of the antenna, the return loss $(S_{11})$ of the antenna with a ground plane was also measured to verify that the performance of the two-element subarray antenna is matched. The measured and simulated results from CST microwave are shown in Figure 72. The actual value used for the resistor on the loaded port is $51\Omega$. For the simulation, $52\Omega$ and $55\Omega$ were used to account for tolerances on the component and PCB trace. The two simulated cases have reasonably good agreement with the measured data.

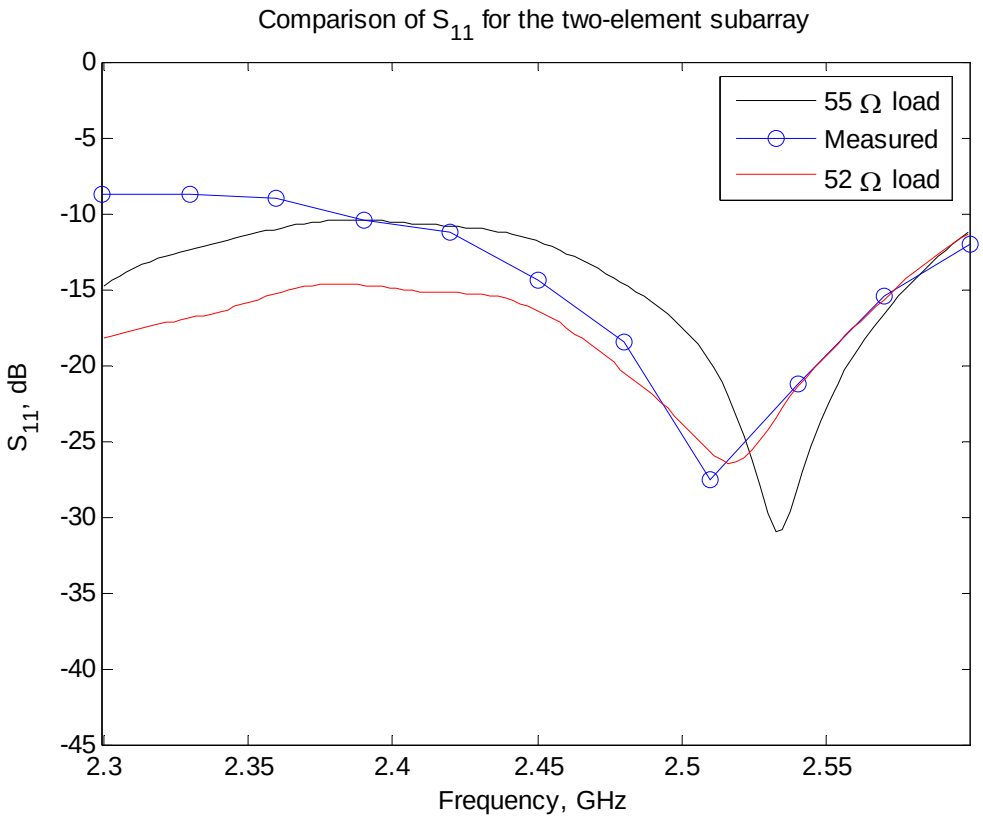



Figure 72. Comparison of measured and simulated return loss of the two-element subarray antenna

## 2. Six-Element Antenna Array Pattern Measurement

The azimuth antenna pattern of the six-element antenna array was also measured in the chamber. The elements were combined in phase using a 1:6 power divider. Note that the array does not operate in this condition, because the beam will be formed digitally. However, this test verifies that there is no unexpected behavior due to mutual coupling or the ground plane. The measured results were plotted using Matlab and the 3 dB beamwidth is estimated to be approximately $20^{\circ}$. From Section 1b, the gain for the two-element subarray was calculated to be around 8.2 dB. For a six-element array, the gain of the array is:

$$\begin{aligned} G_{array} &= G_{sub-array} \times N \\ &= 39.4 \\ &= 16\,\text{dB} \end{aligned} \tag{6.4}$$

where $N$ represents the number of elements in the array. The measured gain is lower than the calculated gain by less than 2 dB and it can be attributed to cable losses from the output of the antenna to the spectrum analyzer used to measure the received power.

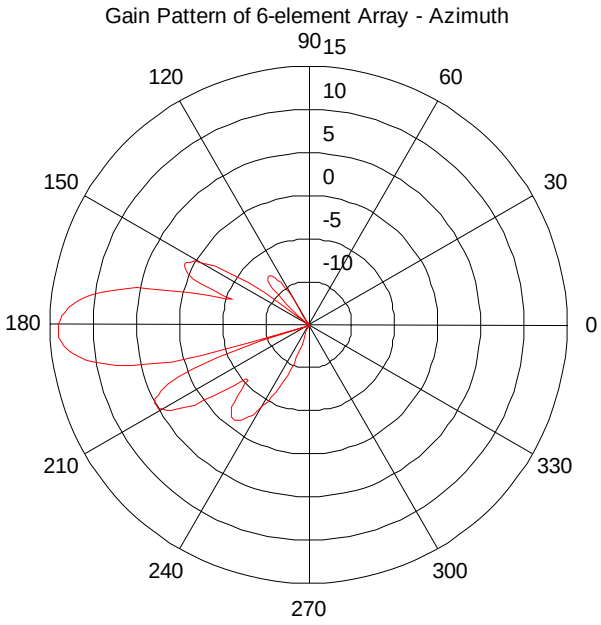



Figure 73. Measured azimuth gain pattern of the six-element antenna array

## B. SOFTWARE MODULES TESTING AND RESULTS

In Chapter V, the development of the three modules—*NTSC Decoding*, *FM Demodulation* and *Tracking Array*—were covered in detail. In this section, the bench top test procedure to validate the software and the results are presented.

### 1. NTSC Decoding Module

To test the NTSC Decoding Module, a video camera was set up with its output fed directly to the 5112 of the PXI. The test was conducted with the video connected to channel 0 of one of the array elements. The antenna elements and demodulator boards were omitted from this test, as this test focused primarily on the ability of the module to receive a video data stream and display it to the terminal. Figure 74 shows the test setup for this module. Figure 75 shows the front panel of the LABVIEW program when running.

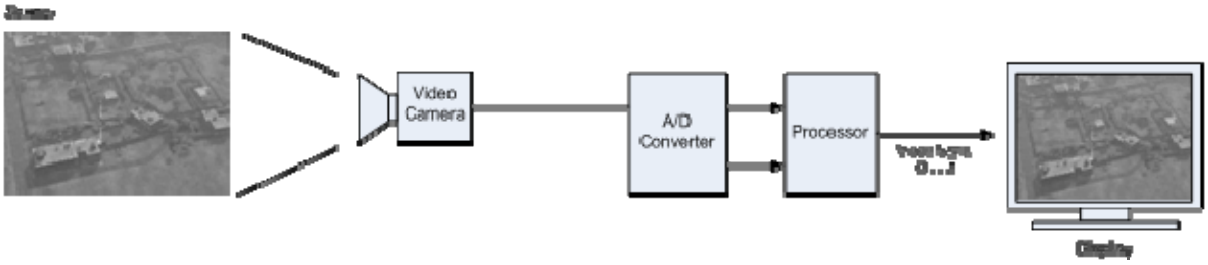



Figure 74. Testing of NTSC Decoding Module

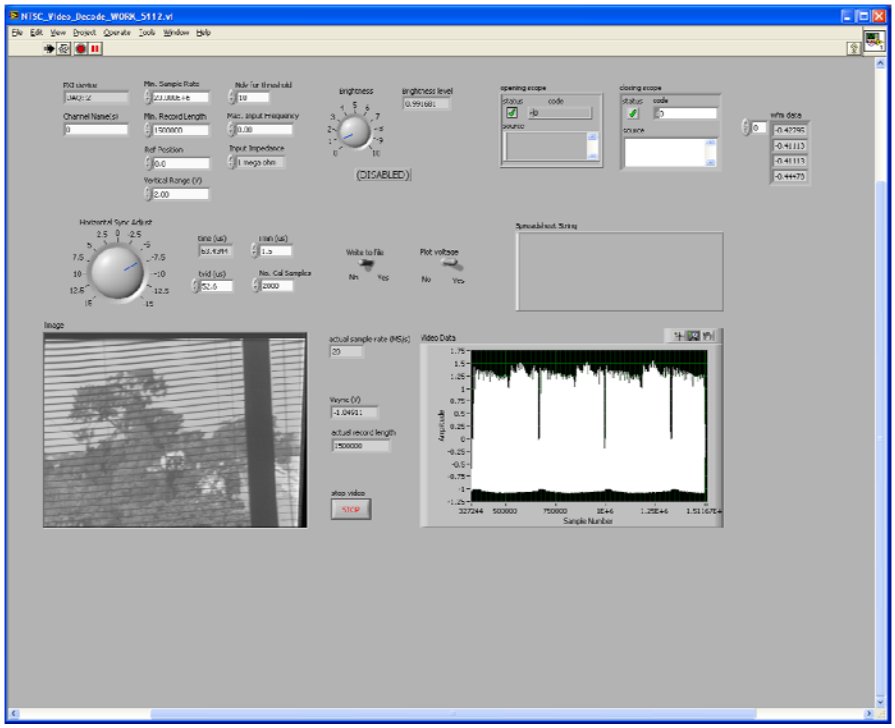
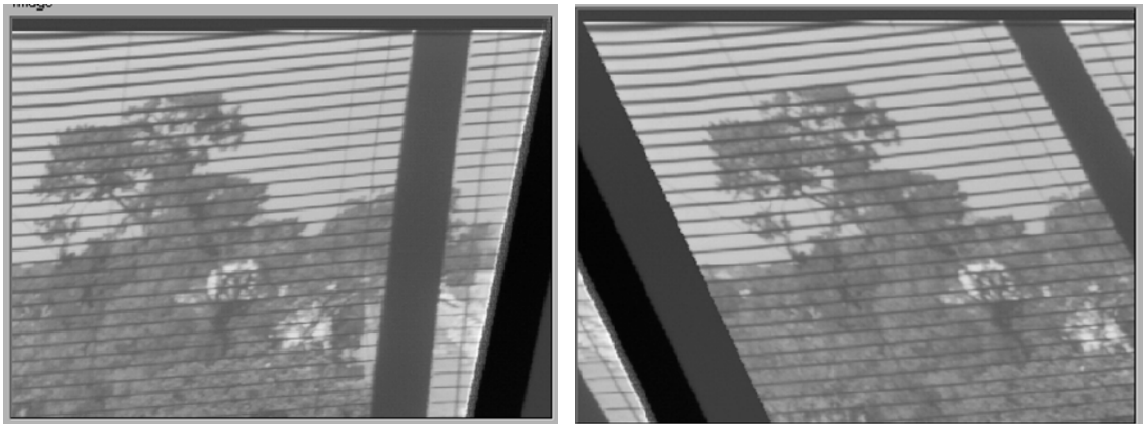


Figure 75. Front panel of NTSC Decoding Module

The setup parameters for the module are similar to the default values used to sample data from the 5112s. The main key component that has to be set for the module is the value of the Horizontal Sync Adjust. When this parameter is set accurately, the video image will be properly displayed. Figure 76 shows the video output in the event this parameter is inaccurately set.



(a) (b)

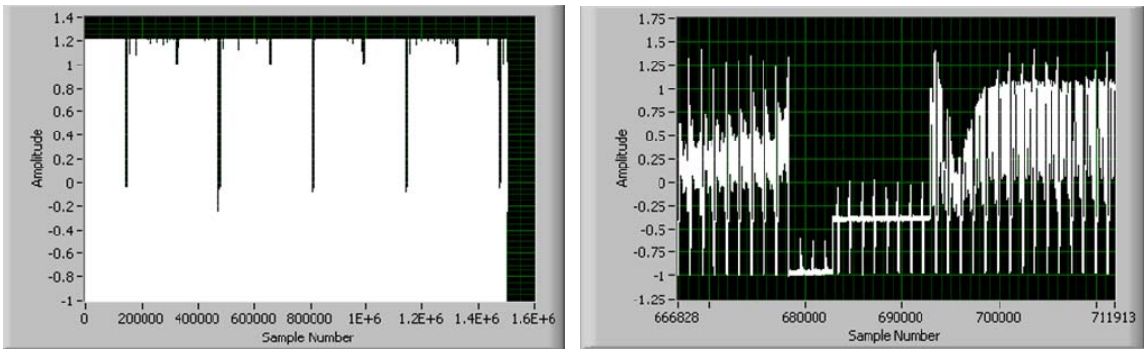
Figure 76. Variation of Horizontal Sync Adjust setting: (a) Lower value, (b) Higher value

The output of the video image from the module was compared against the location that was being captured by the video camera. For this purpose, a digital camera was used to obtain a snapshot of the same area to compare the video output of the module. Figure 77 shows the comparison of the video output of the module and the still image of the same scene captured using a digital camera. It is observed that the module is able to accurately display the video image. The lower quality of the video is primarily due to having only one sample per pixel. This allows a faster frame processing time.

(a) (b)

Figure 77. Comparison of the video output from (a) NTSC Decoding Module, and (b) a snapshot taken using a digital camera of the same location in focus

By analyzing the video data plot, several features of the signal that was covered in Chapter II can be observed. Figure 78 shows two plots of the video data stream: (a) is a macro view of the different fields received in the video stream, and (b) shows a expanded view of one of the regions. In (a), there are several regions of 'vertical blanks' in the plot. These represent the sync signal information that precedes the tracing of a new line in the video frame. From (b), the horizontal sync signal (HSYNC), which signals the beginning of a new line, and back the porch of the signal can be easily identified.




(a) (b)

Figure 78. Plot of video data stream received by the 5112

## 2. FM Demodulation Module

To test the FM Demodulation Module, a simple setup consisting of a signal generator, signal synthesizer, AD8347 demodulator boards and the National Instruments PXI was assembled for the testing. The test setup is shown in Figure 79

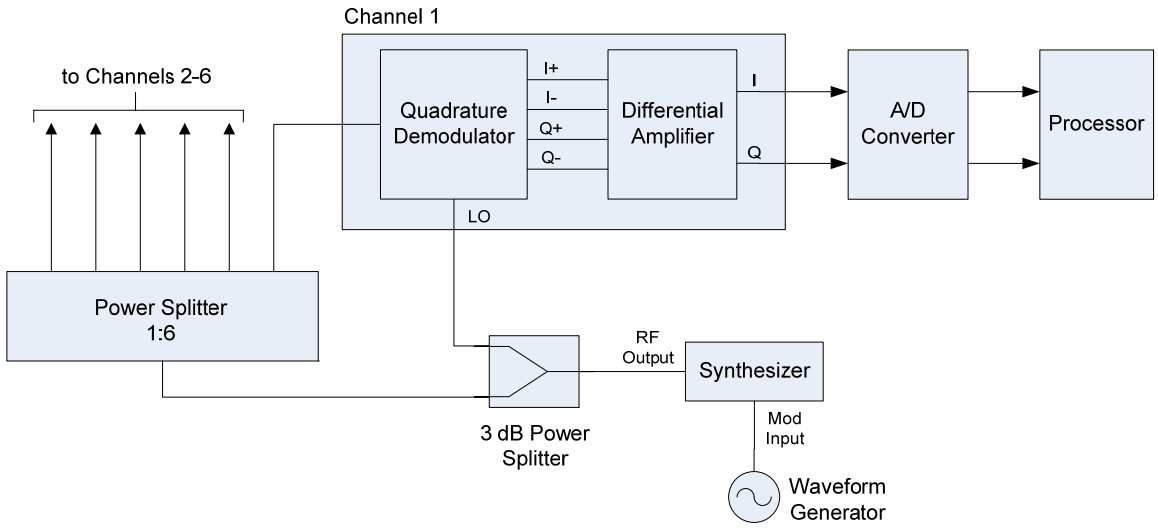



Figure 79. Testing of FM Demodulation Module

The module was tested with a sine wave input at a frequency of 50 kHz and amplitude of 1V peak to peak. The signal generator was used to generate the wave. The

sine wave was then fed to the signal synthesizer, which modulated the sine wave using FM, upconverting the signal to RF at 2.4 GHz. This simulates the transmission of an FM modulated signal from a UAV to the array antenna. This signal was then fed to the AD8347 demodulator boards to down convert the signal to baseband. Practically, this signal would have been received by the antenna elements before it was fed to the demodulator boards. For this lab test, the antenna elements were not included in the test. The *I* and *Q* components from the demodulator boards were then connected to the 5112 board and sampled in the digital domain. Figure 80 shows the front panel of the FM demodulation software for the sine wave input. Figure 81 and Figure 82 show the expanded displays from the front panel.

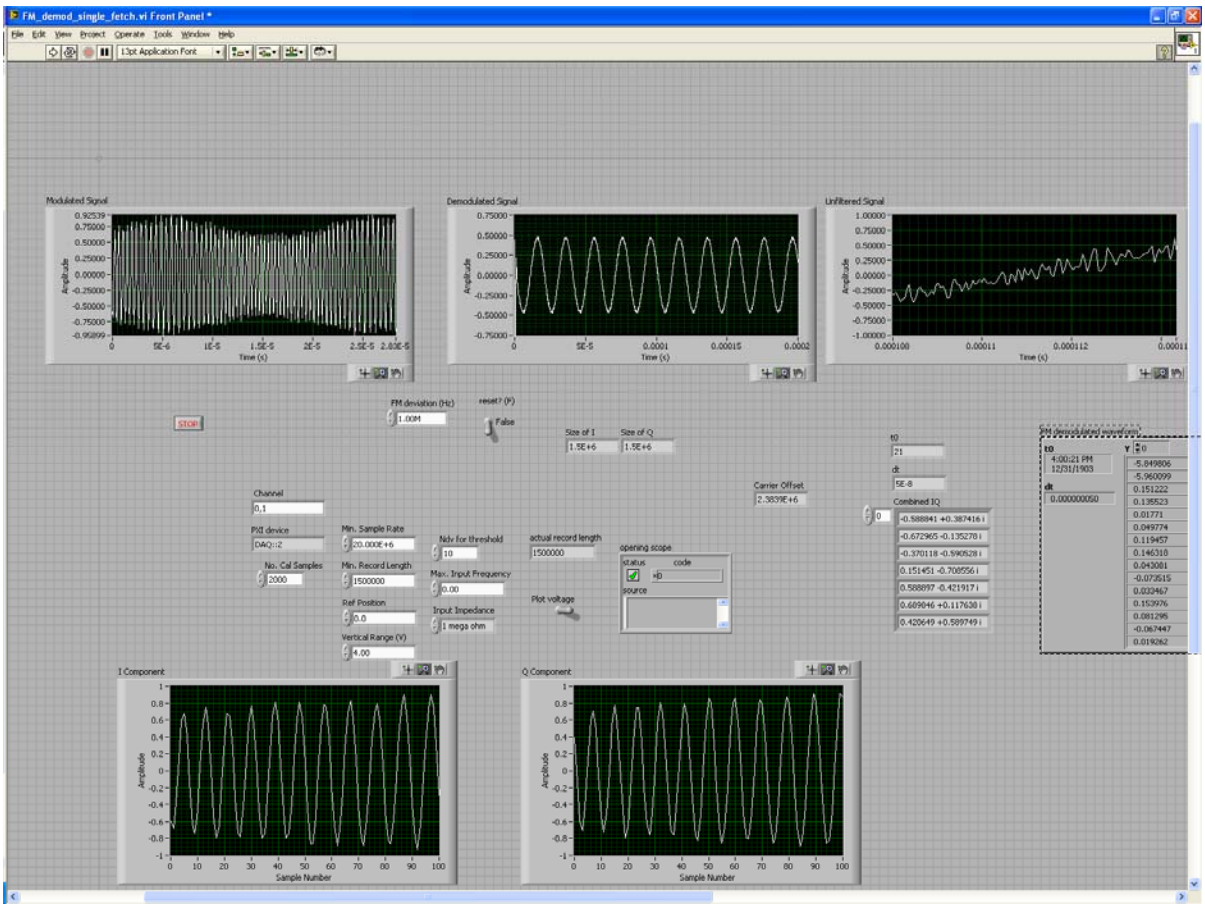



Figure 80. Front panel of the FM demodulation software for a sine wave input

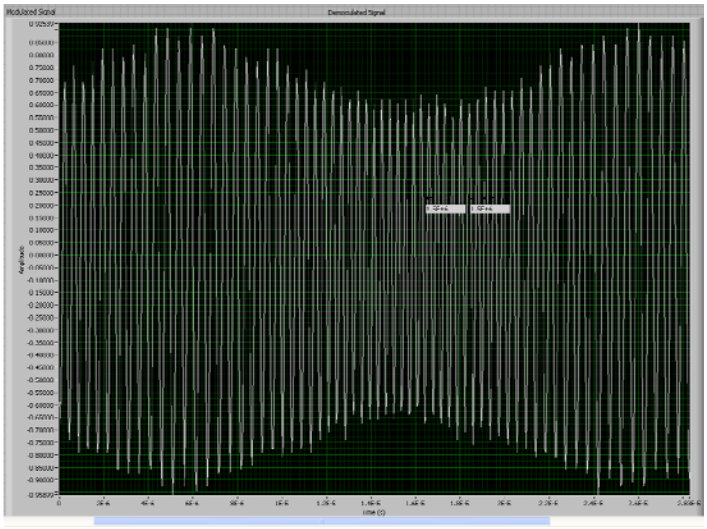


Figure 81. FM modulated sine wave at baseband

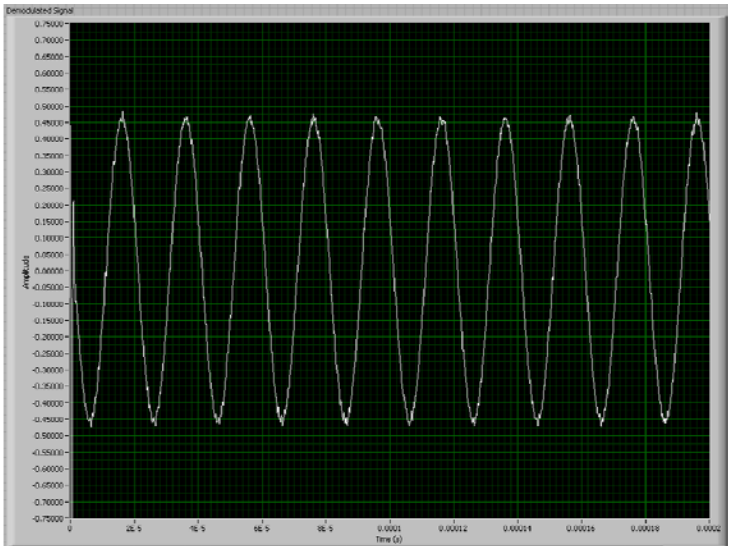


Figure 82. Demodulated sine wave recovered after filtering

When an input sine wave was used, the original waveform was retrieved after the filter. Based on Nyquist, the filter cutoff frequency for this test was set at twice the message frequency (50 kHz x 2 = 100 kHz). It was observed that when the input signal was decreased to below 20 kHz, the amplitude of the waveform decreased and the signal could not be adequately retrieved. At lower frequencies, the amplitude of the waveform was observed to decrease until the sinusoidal waveform could no longer be distinguished.

When the input waveform was changed from a sine wave to a square wave, triangle wave and ramp wave, the module failed to accurately retrieve the input signal.

Figure 83 shows the output of the module for an input square waveform. On the input *Q* component plot, a square wave is observed. This is not the case for the *I* component plot. The demodulated signal retrieved represented a sinusoidal waveform with the same characteristic frequency as the input waveform. The fact that only the fundamental waveform was retrieved meant that higher frequency components of the waveform, used to characterize square waves, were filtered out. Tests were conducted with various filter bandwidths but the square wave could not be retrieved.

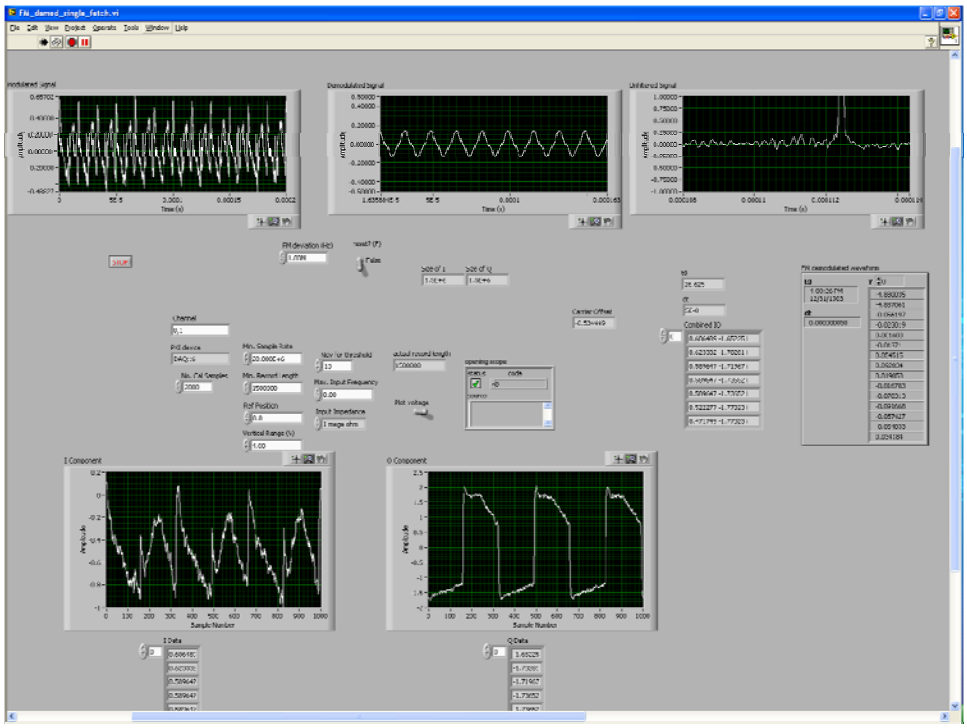


Figure 83. Output of FM Demodulation Module for a square wave input

Figure 84 and Figure 85 shows the output of the module for triangle wave and ramp wave inputs. For both these cases, the *I* and *Q* component plots, had representative plots to the input. The output waveform, however, did not represent the input waveform. For the case with triangle wave input, both the *I* and *Q* components resembled a triangle wave. When these were combined to form the complex envelope of the signal, the

triangle wave was still evident. The output, however, could only characterize the fundamental frequency of the input wave. For the case with the saw tooth wave input, there was an observed inversion of graphs between the *I* and *Q* components. This module did not produce the expected results and will have to be studied further in future work.

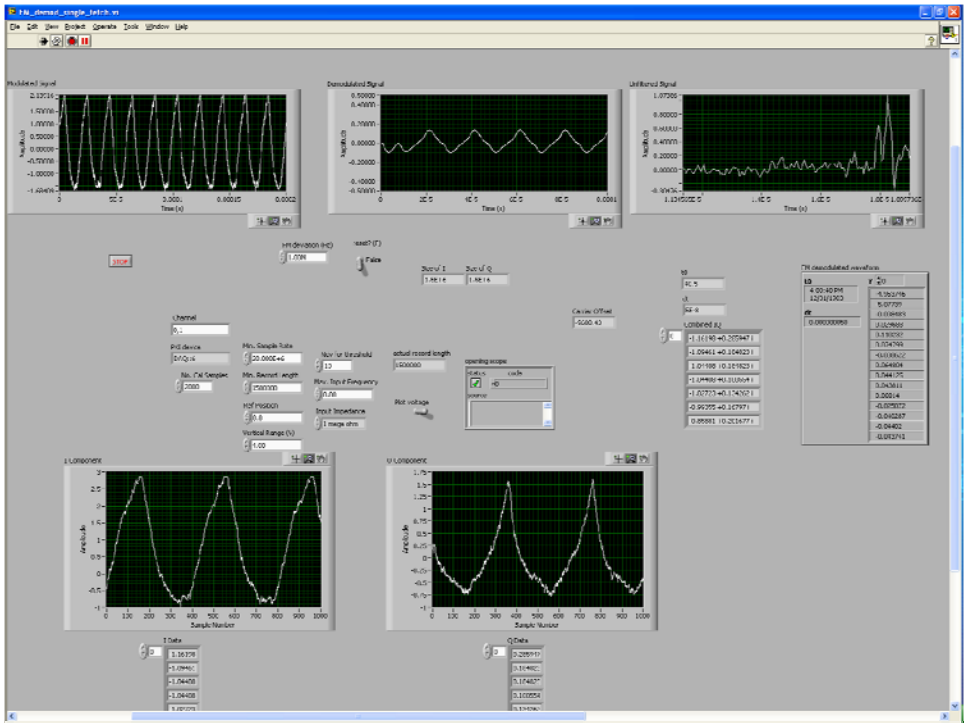


Figure 84. Output of FM Demodulation Module for an triangle wave input

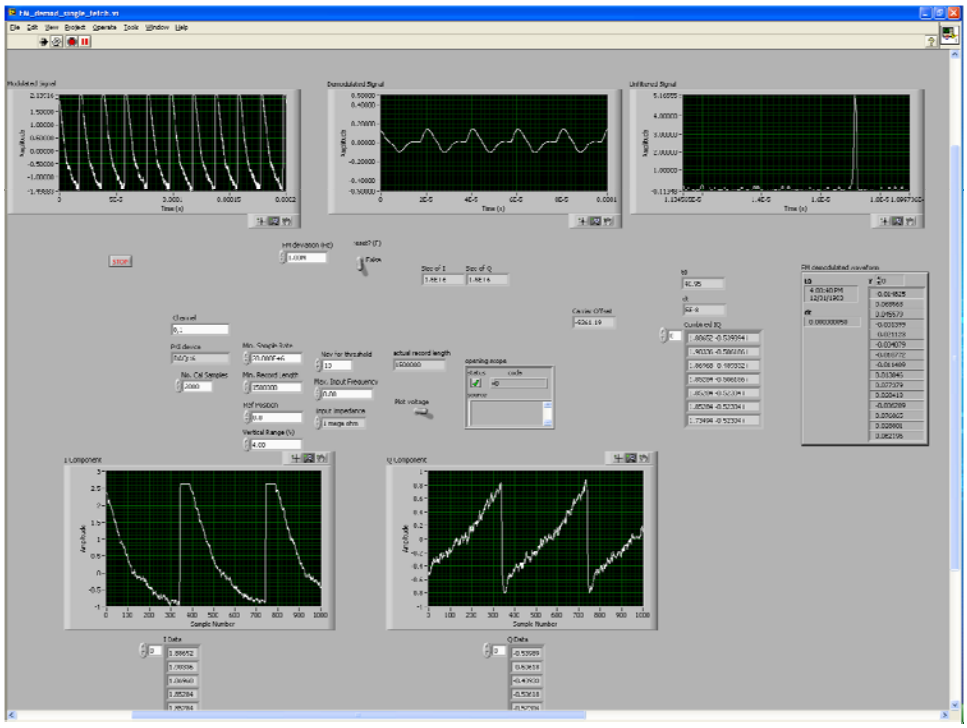


Figure 85. Output of FM Demodulation Module for a saw-tooth wave input

### 3. Tracking Array Module

Developing a bench top test plan for the Tracking Array Module proved to be a non-trivial task. For this test, a lab setup was required to simulate the arrival of plane waves from a source in the far field to determine if the module was capable of correctly acquiring the waveform's angle of arrival. For the lab test of the module, a signal synthesizer, the six-element antenna array and the PXI was used. Similar to the test conducted for the FM Demodulation Module, the antenna elements were not used and instead a signal tone was fed directly to the RF input of the demodulator boards. Phase shifters were introduced at the input of the demodulator boards to simulate a plane wave arriving from an angle off of boresight. As discussed in Chapter V, there are two methods of tracking the intended target: RSNS and RSNS followed by monopulse DBF. The module developed here uses a modified version of the RSNS routine developed in [10]. Since the routine had been validated and found to work well, this lab test focused on tracking using DBF. Figure 86 shows the test setup for the module.

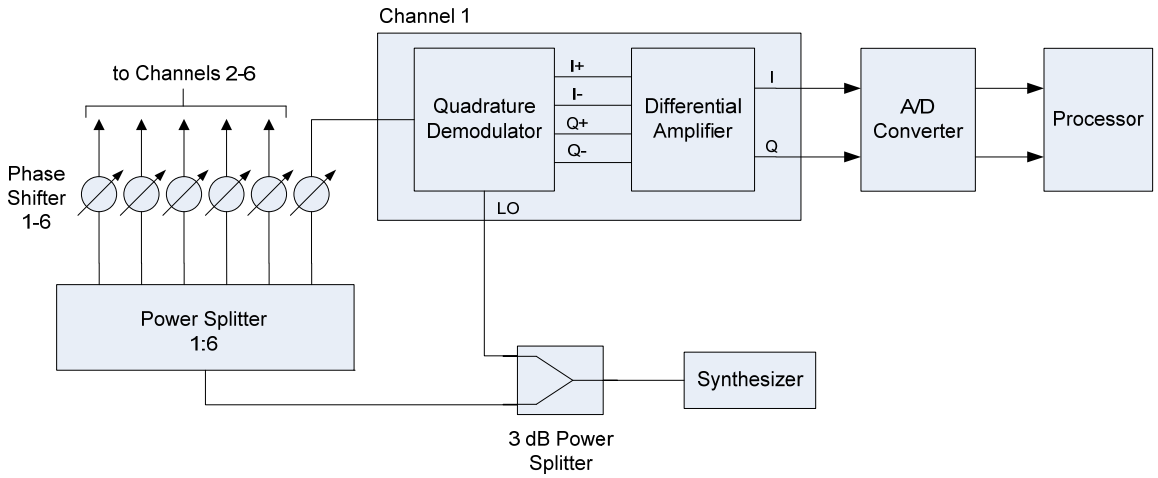



Figure 86. Testing of Tracking Array Module.

The signal synthesizer was used to generate a signal tone at 2.4 GHz and this signal was fed to the phase shifters at the input of each of the demodulator boards. The phase shifters introduced a phase shift at each element, which corresponded to a specific angle of arrival, $\theta$, computed for a fixed frequency and element spacing based on Equation (3.17). Due to the limitation of achieving fine resolution of phase difference between the antenna elements, the test was conducted only for three discrete values of $\theta$: $-10^{\circ}, 0^{\circ}$ and $10^{\circ}$. This was deemed sufficient as an initial test, prior to a comprehensive test in the anechoic chamber, to ensure the module works well for both positive and negative values of $\theta$. Table 6 shows the output of the tracking array module for the three different values of $\theta$.

Table 6. Output of Tracking Array Module using the phase shifters to simulate three angles of arrival

| Angle of Arrival/$\theta$ | Phase Difference Between Elements, $\phi$ | Computed Angle of Arrival in Module |
|---|---|---|
| $-10^{\circ}$ | 32 | -9.4 |
| $0^{\circ}$ | 0 | 0 |
| $10^{\circ}$ | -32 | 9.6 |

From the results shown in Table 6, it can be seen that the array with monopulse DBF was able to successfully determine the angle of arrival of the incoming signal based on the phase information obtained by each of the array elements. Although this test only handles three values of $\theta$, it gave a good indication that the DBF logic was properly implemented and the module could then be integrated with the antenna elements for complete testing in the anechoic chamber.

## C. ACQUISITION/TRACKING TESTING IN THE ANECHOIC CHAMBER

Once the individual component testing was completed, the next phase of the project involved the integration of the hardware and software components into a complete antenna array. The test was conducted in an anechoic chamber so as to minimize errors due to reflections when creating a test environment to simulate incident plane waves. Within the chamber, the antenna array was set up on a pedestal. The pedestal is controlled by equipment outside of the chamber, which allows the antenna array to be rotated to different angles from the designated boresight. A horn antenna was setup as a transmitter on the opposite end of the array within the chamber (see Figure 66). The distance between the horn antenna was 19 feet (5.8 m). At this distance, at the operating frequency of 2.4 GHz, the array is within the far field of the transmitted wave from the antenna. Figure 87 shows the equipment setup in the anechoic chamber. Figure 88 and Figure 89

shows the setup of the transmitter antenna array in the anechoic chamber. Figure 90 shows the equipment used to control the pedestal and the signal transmitter, within the chamber.

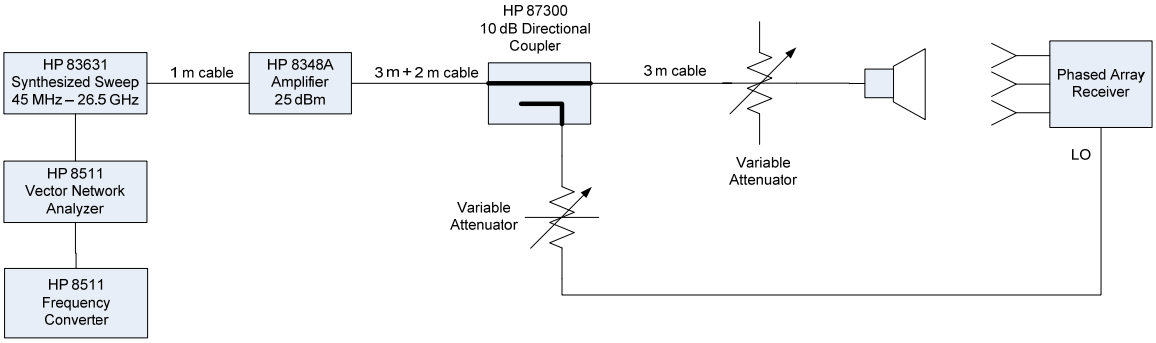



Figure 87. Test setup in anechoic chamber

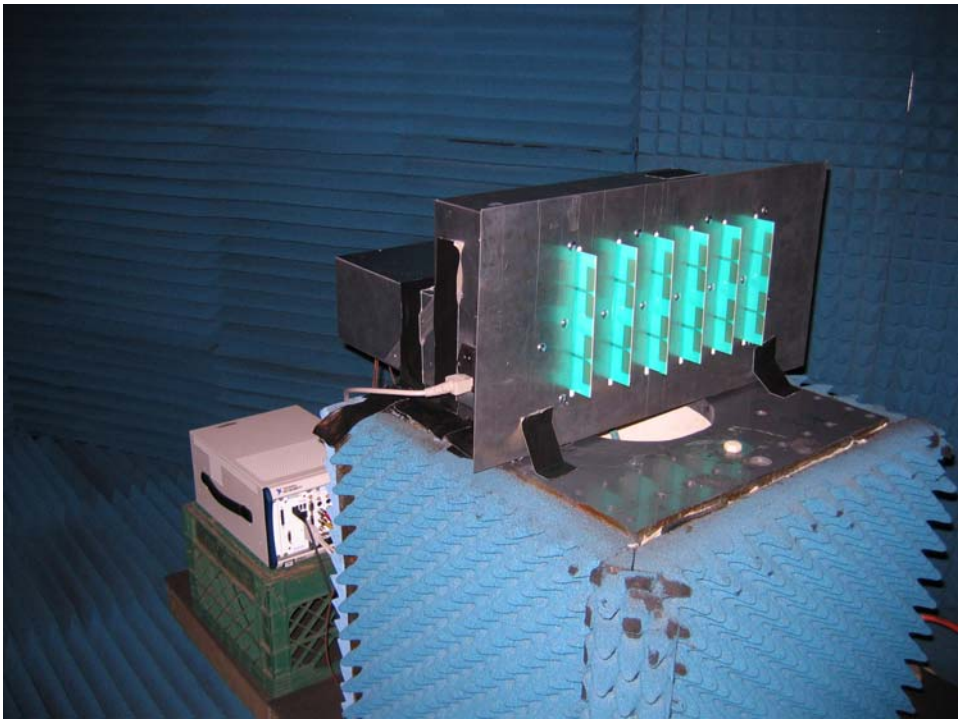


Figure 88. Setup of Array on a pedestal in anechoic chamber

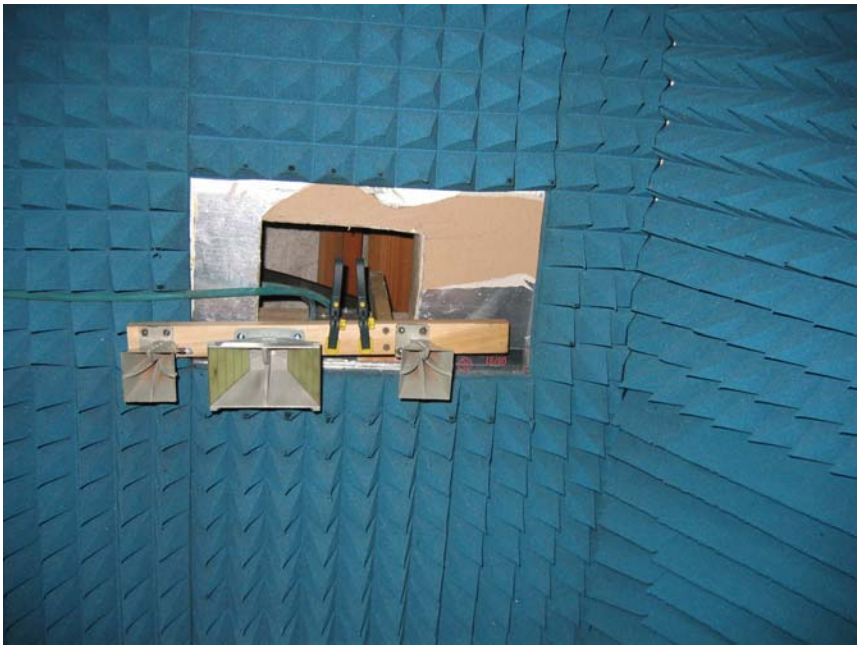


Figure 89. Transmitter used in the anechoic chamber

Figure 90. Equipment used to control the pedestal within the chamber

### 1. Initialization

Prior to running the test sequence in the chamber, there are several initialization procedures that were executed to ensure accurate results. One key aspect of the initialization procedure is obtaining the offsets for the demodulator boards. These offsets were obtained for the bench top test conducted previously for the hardware and thus there was no requirement to repeat this procedure. There is also a requirement to obtain the phase adjustments for each array element at boresight. This requirement was discussed in Chapter V. This was done by setting the pedestal to $0^{\circ}$ and obtaining the phase values of each element. These values were then used in the module as phase adjustments for the respective elements so that the phase difference between elements is approximately $0^{\circ}$ when the incident wave arrives from boresight.

### 2. RSNS/DBF Acquisition/Tracking Test for Discrete AOA

With all the initialization procedures completed, the RSNS/DBF acquisition tracking of the system was tested. This was done by setting the angle of arrival (AOA), at distinct steps from $-30^{\circ}$ to $30^{\circ}$, in increments of $5^{\circ}$ and observing the scan angle computed by the antenna array. The AOA was changed by making an equivalent change to the pedestal angle. The scan angle for each case was plotted against the fetch number for each record obtained from the 5112. Sampling is done at 20 MHz, with each record consisting of 150,000 samples. The phase value for each element was obtained for each case and a MATLAB program was written to derive the AOA based on this phase information. This is represented as Computed AOA in the plots. In addition to the plot stated above, the correction angle computed for each iteration by the *Tracking Array Module* was also plotted against fetch number. Figure 91 to Figure 116 show the plots for the scan angle and correction angle against the fetch number for each of the cases.

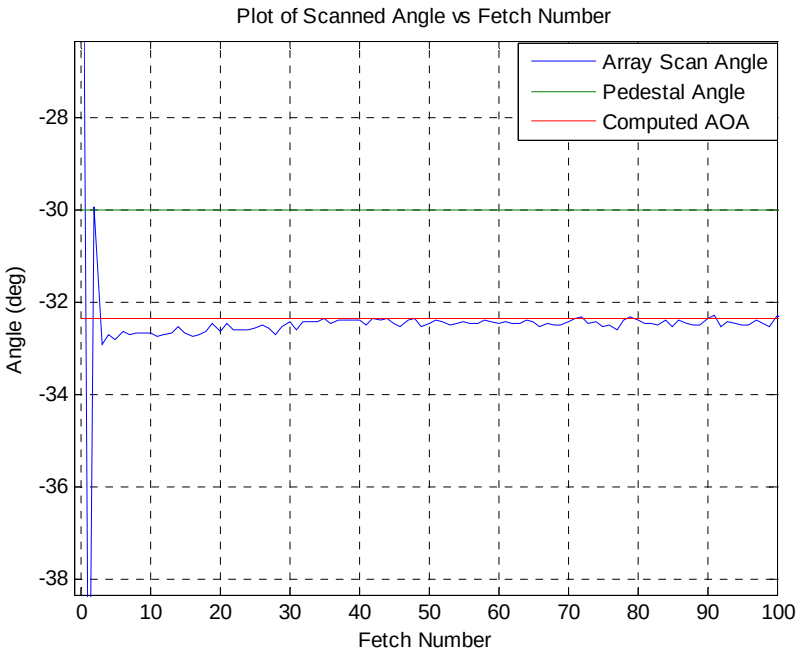



Figure 91. Plot of Angle against Fetch Number for AOA of $-30^{\circ}$

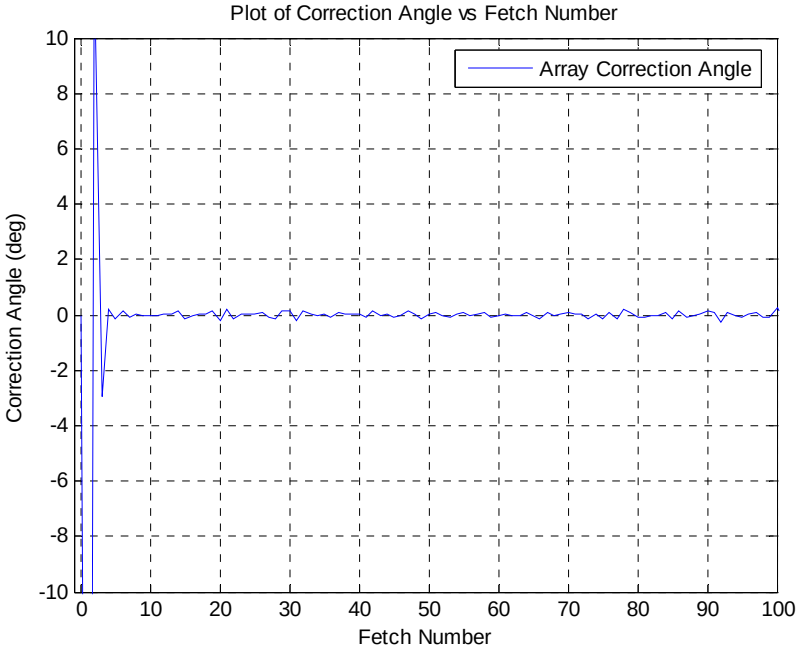



Figure 92. Plot of Correction Angle against Fetch Number for AOA of $-30^{\circ}$

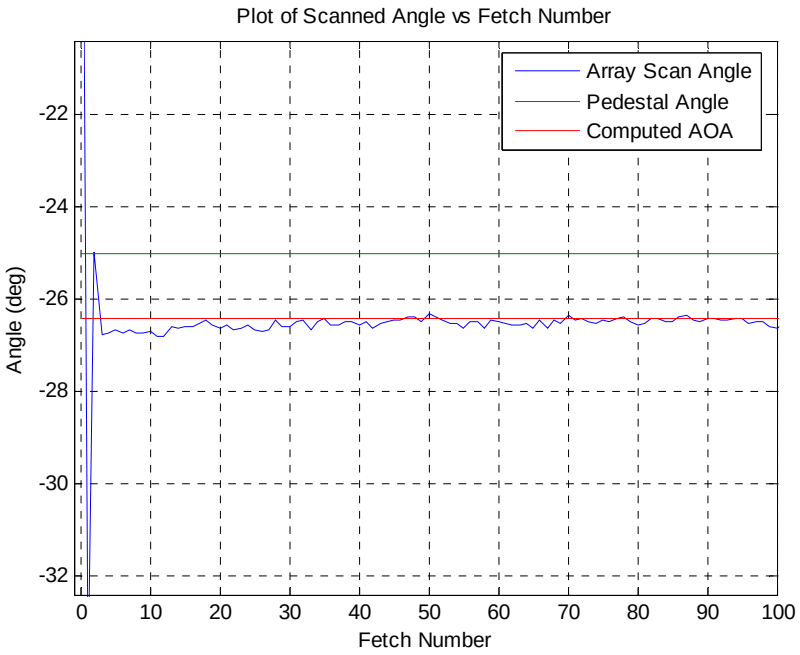



Figure 93. Plot of Angle against Fetch Number for AOA of $-25^{\circ}$

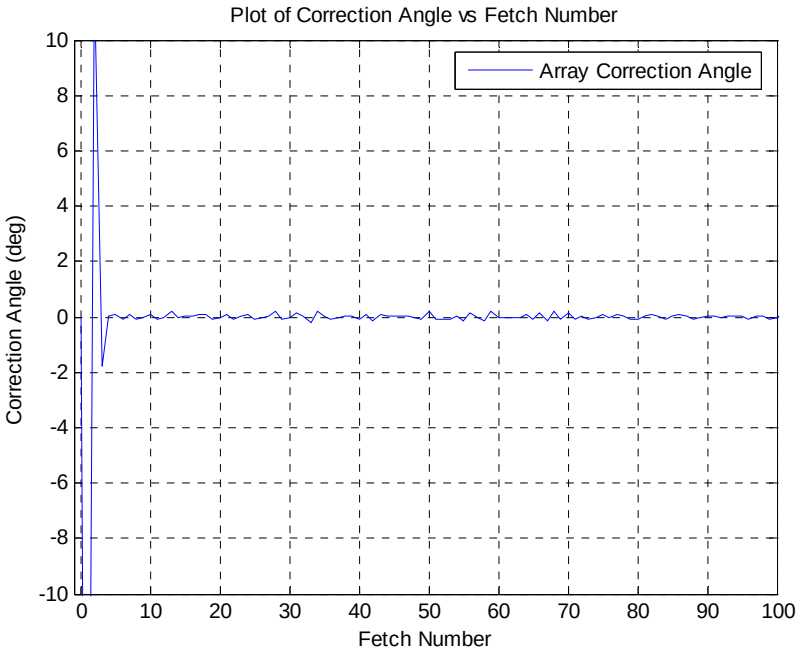



Figure 94. Plot of Correction Angle against Fetch Number for AOA of $-25^{\circ}$

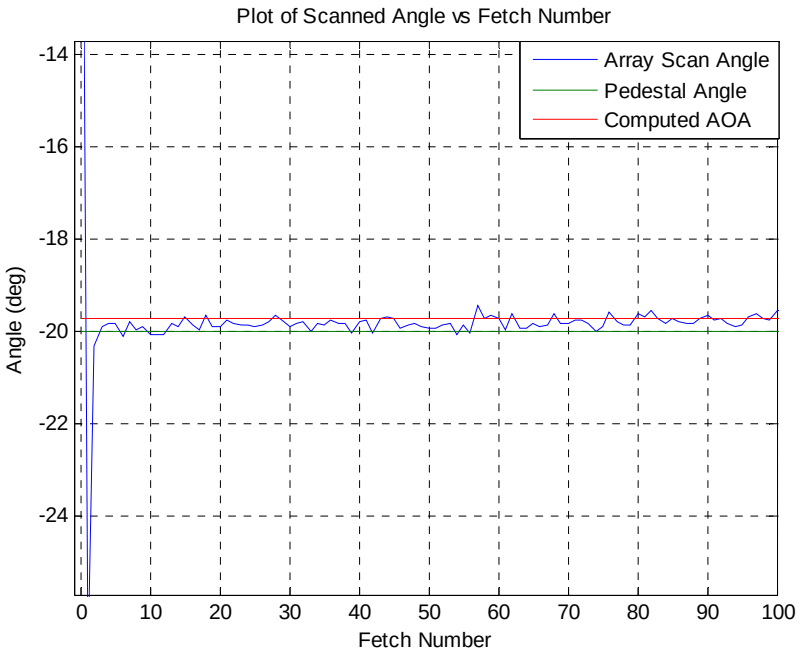



Figure 95. Plot of Angle against Fetch Number for AOA of $-20^{\circ}$

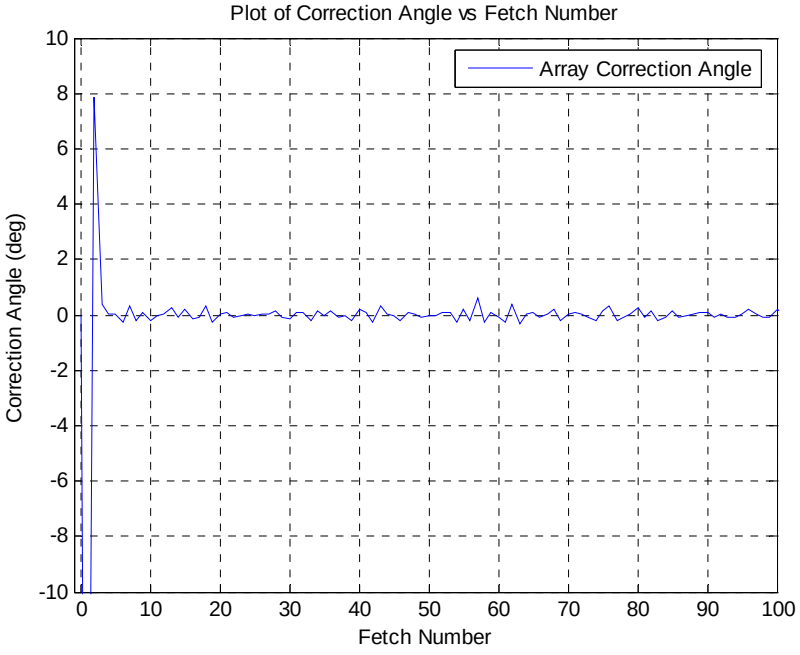



Figure 96. Plot of Correction Angle against Fetch Number for AOA of $-20^{\circ}$

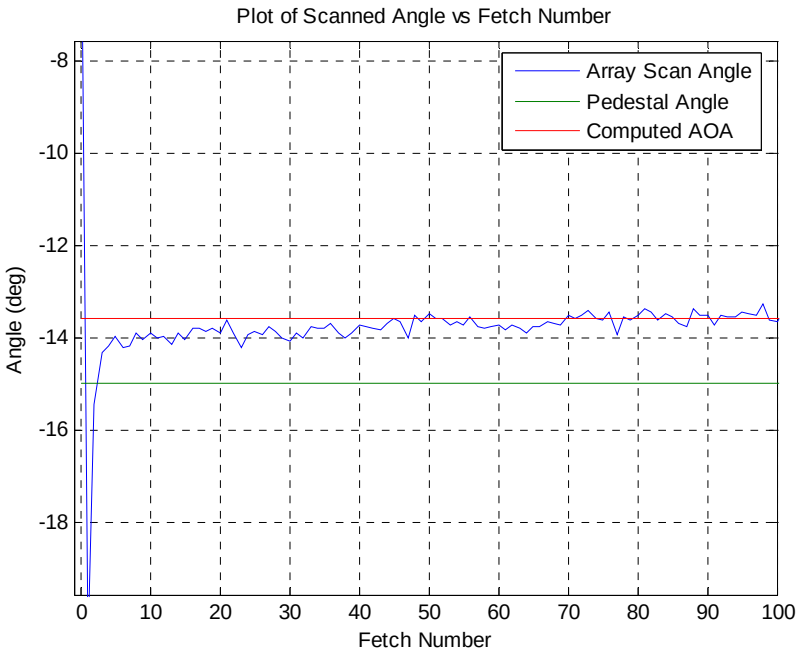



Figure 97. Plot of Angle against Fetch Number for AOA of $-15^{\circ}$

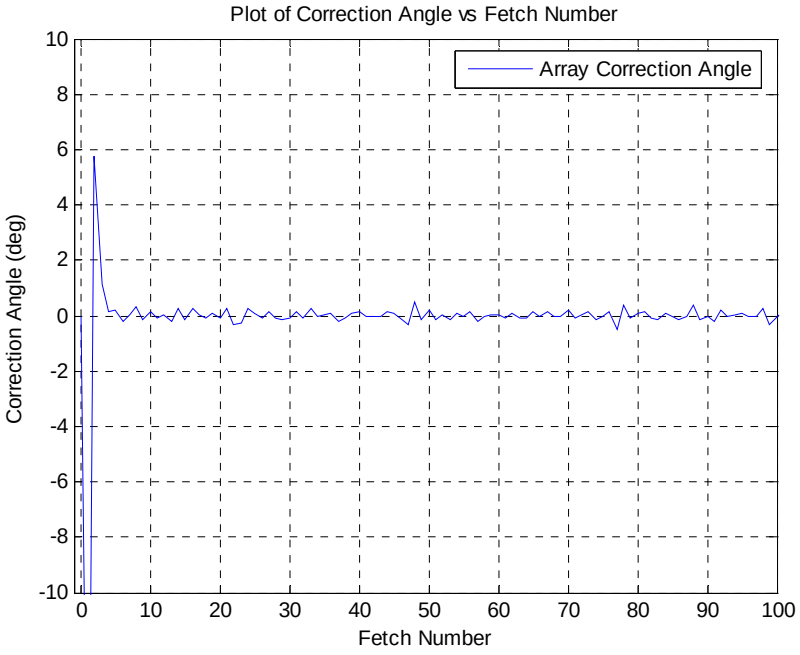



Figure 98. Plot of Correction Angle against Fetch Number for AOA of $-15^{\circ}$

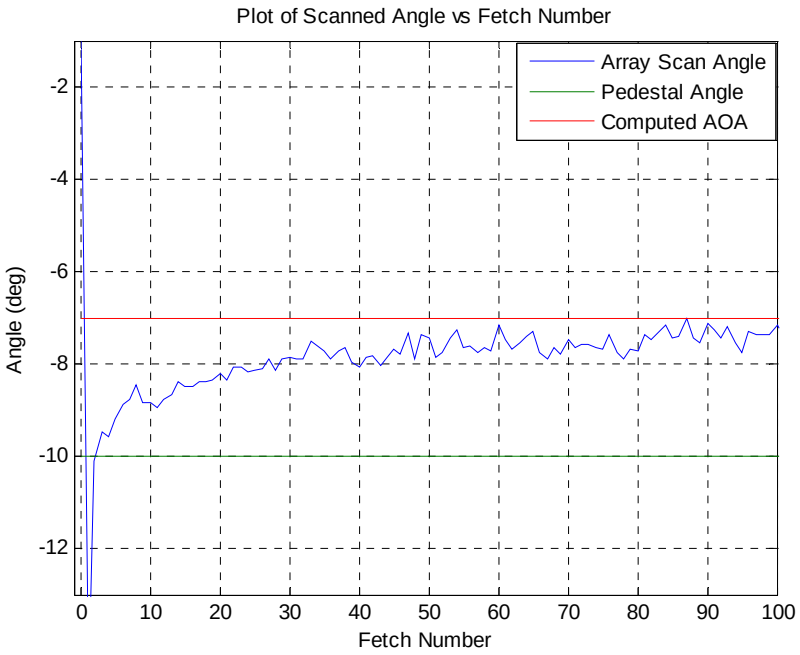



Figure 99. Plot of Angle against Fetch Number for AOA of $-10^{\circ}$

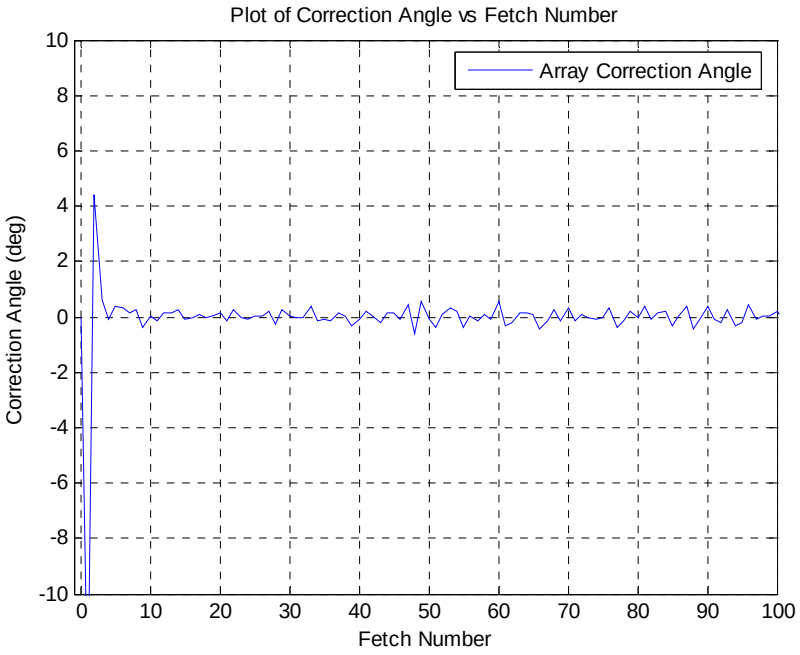



Figure 100. Plot of Correction Angle against Fetch Number for AOA of $-10^{\circ}$

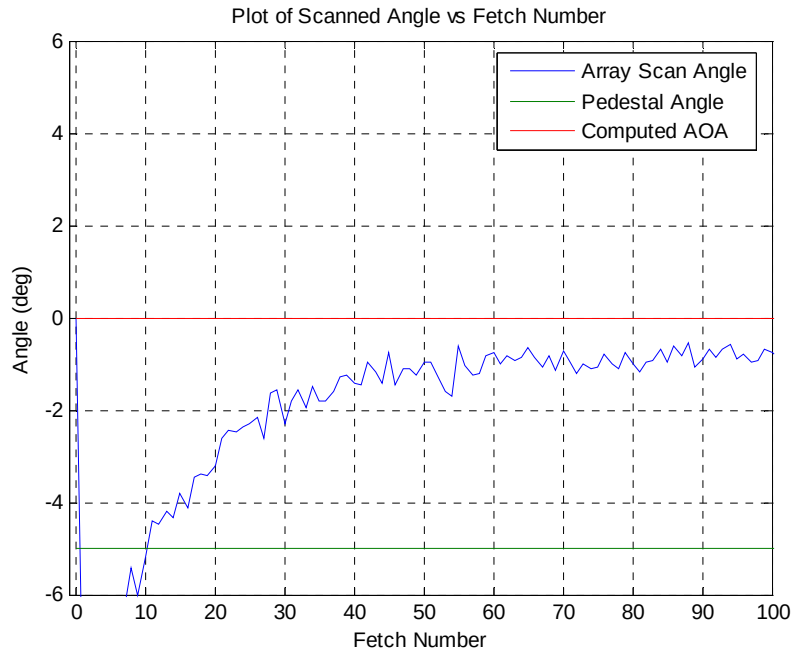



Figure 101. Plot of Angle against Fetch Number for AOA of $-5^{\circ}$

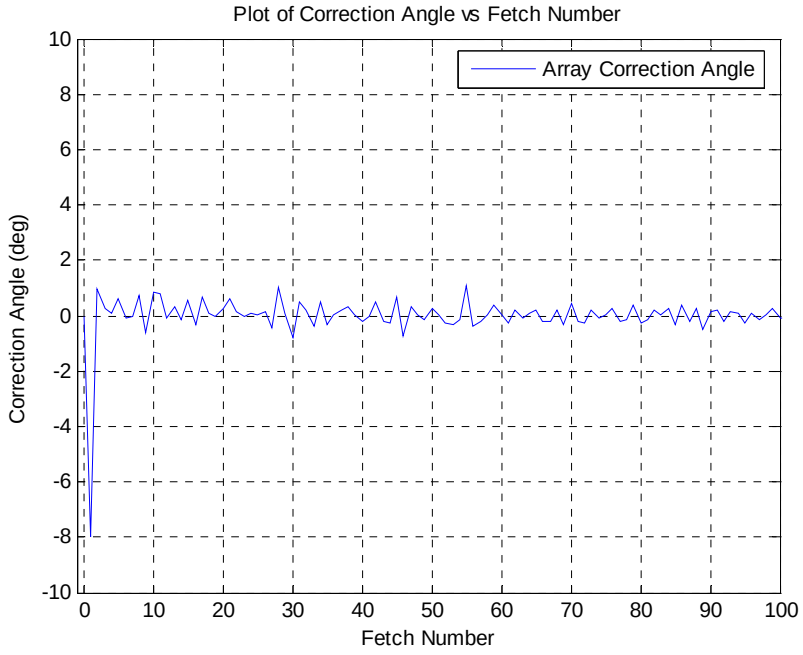



Figure 102. Plot of Correction Angle against Fetch Number for AOA of $-5^{\circ}$

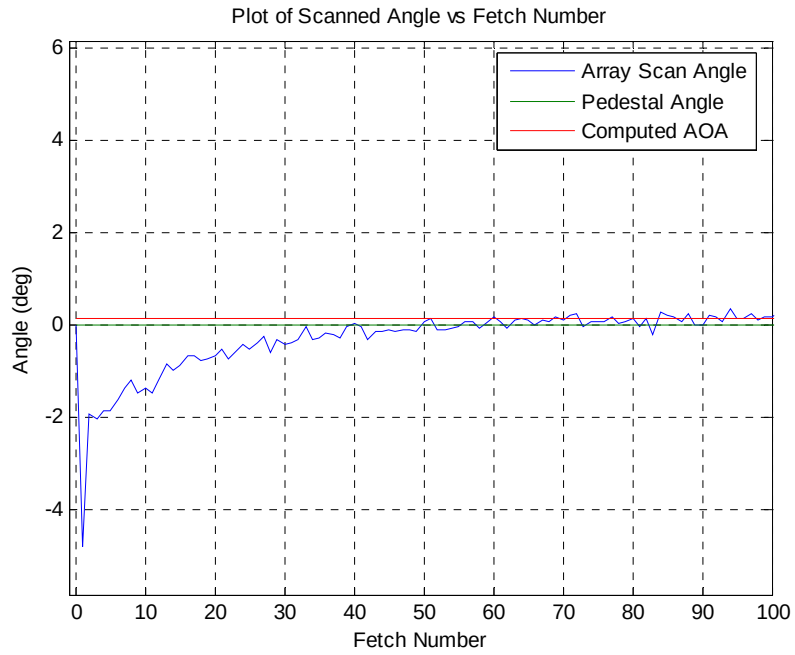



Figure 103. Plot of Angle against Fetch Number for AOA of 0°

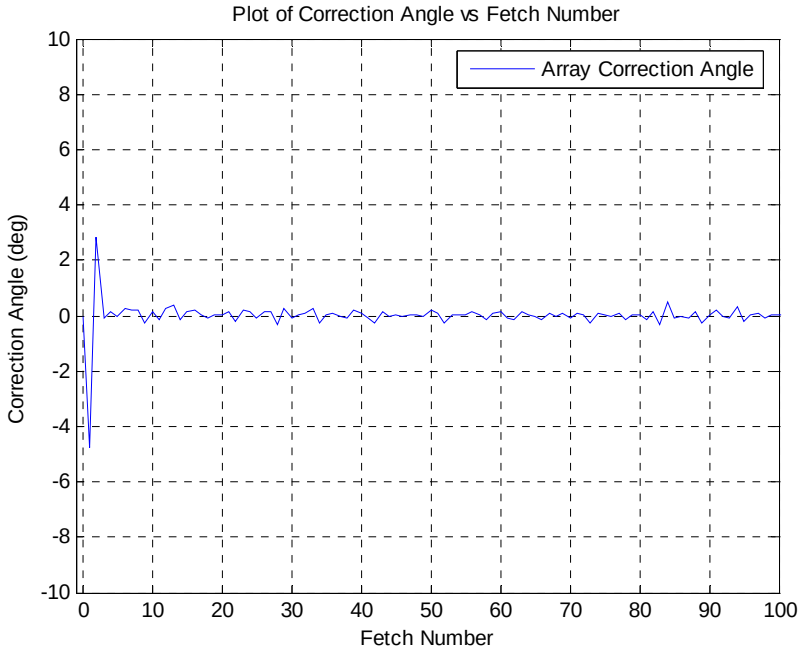



Figure 104. Plot of Correction Angle against Fetch Number for AOA of 0°

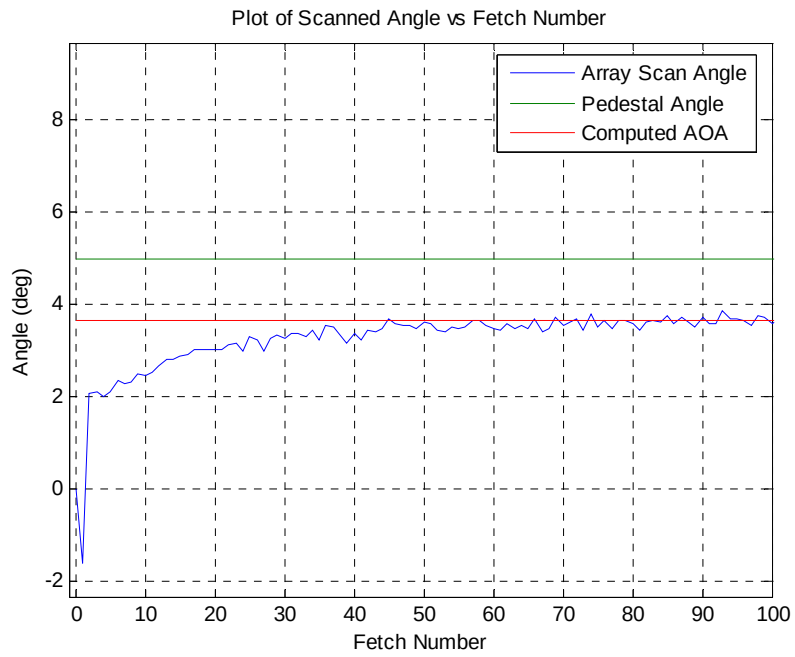



Figure 105. Plot of Angle against Fetch Number for AOA of 5°

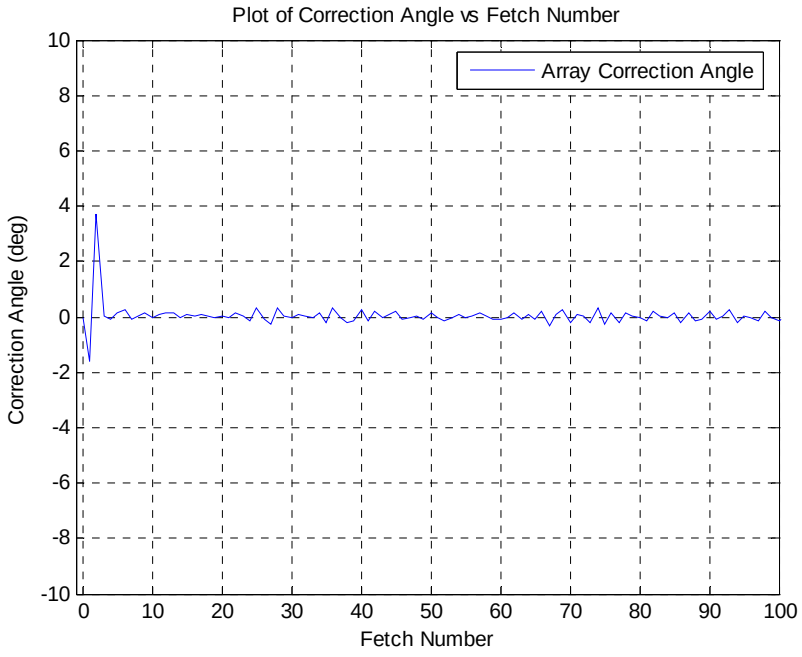



Figure 106. Plot of Correction Angle against Fetch Number for AOA of 5°

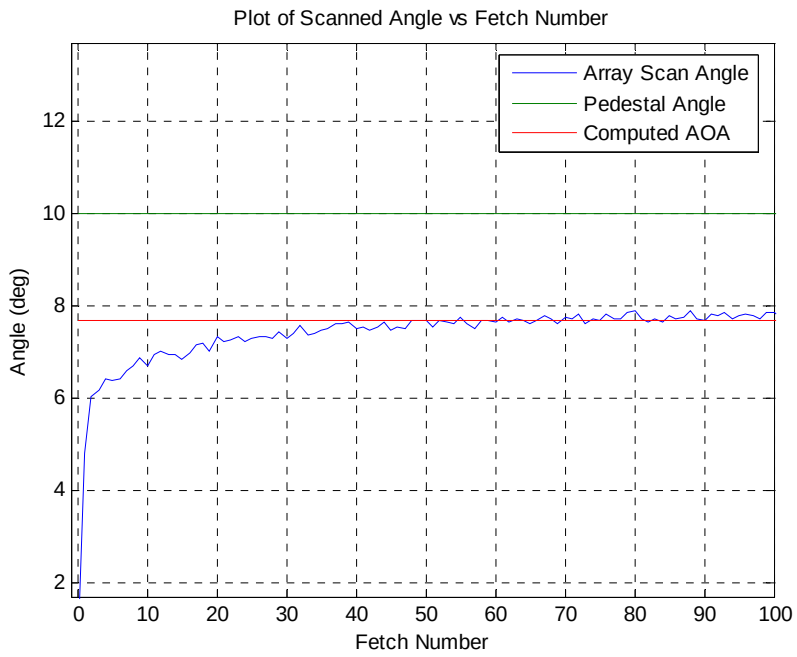



Figure 107. Plot of Angle against Fetch Number for AOA of 10°

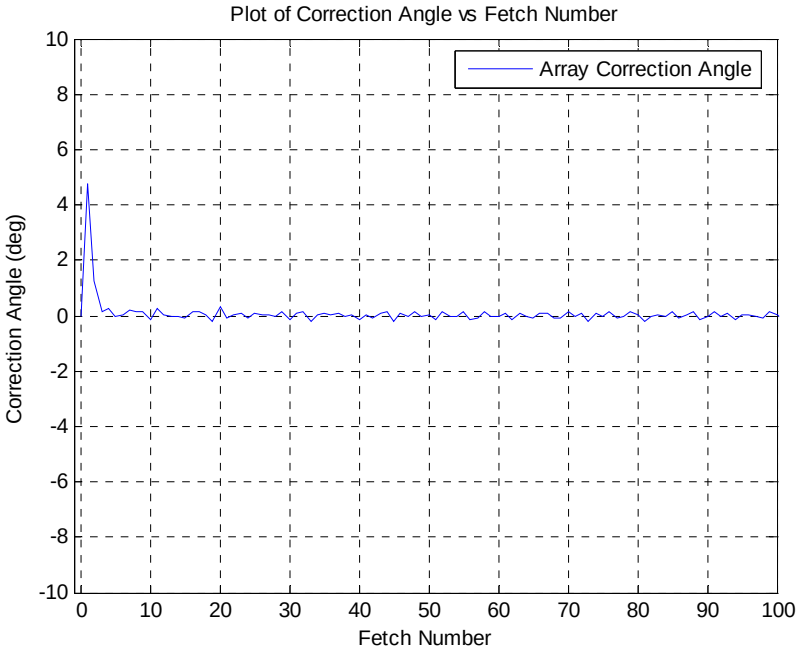



Figure 108. Plot of Correction Angle against Fetch Number for AOA of 10°

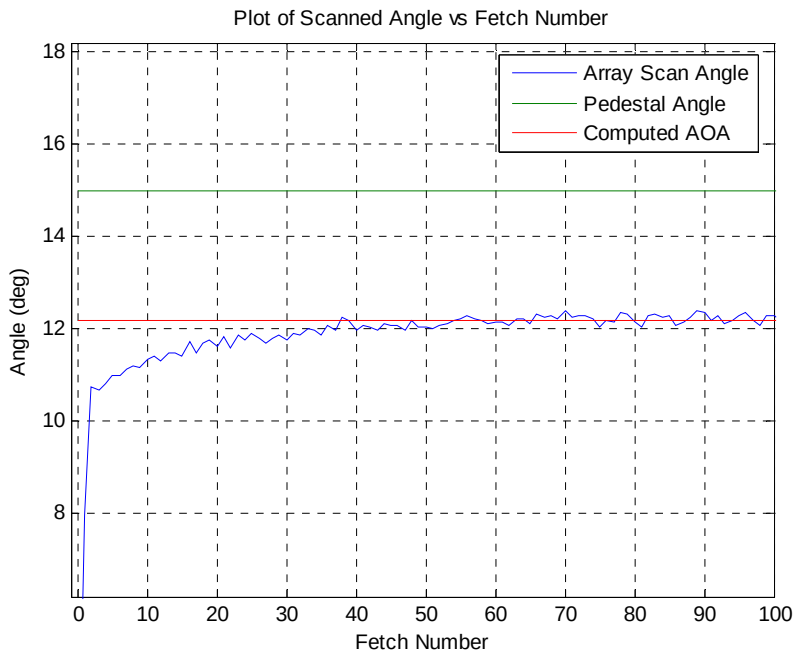



Figure 109. Plot of Angle against Fetch Number for AOA of 15°

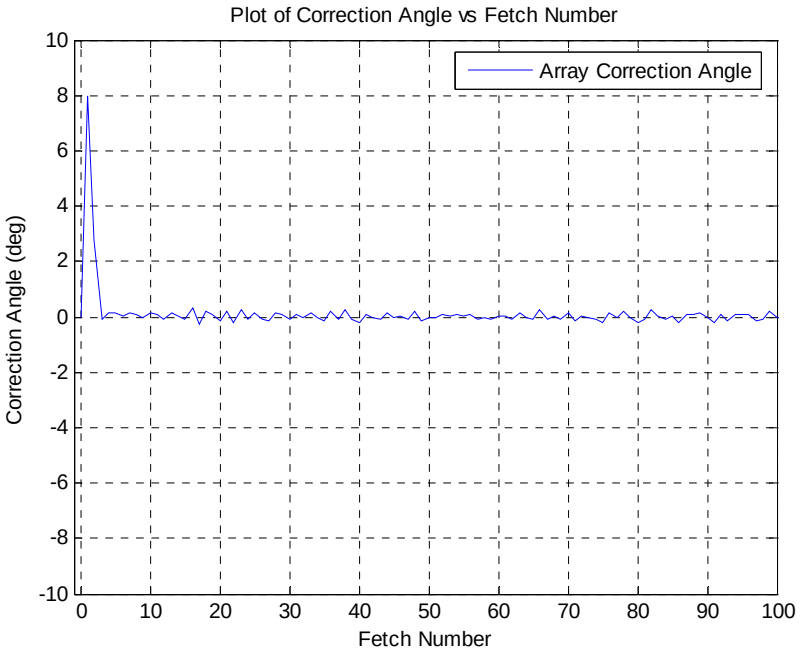



Figure 110. Plot of Correction Angle against Fetch Number for AOA of 15°

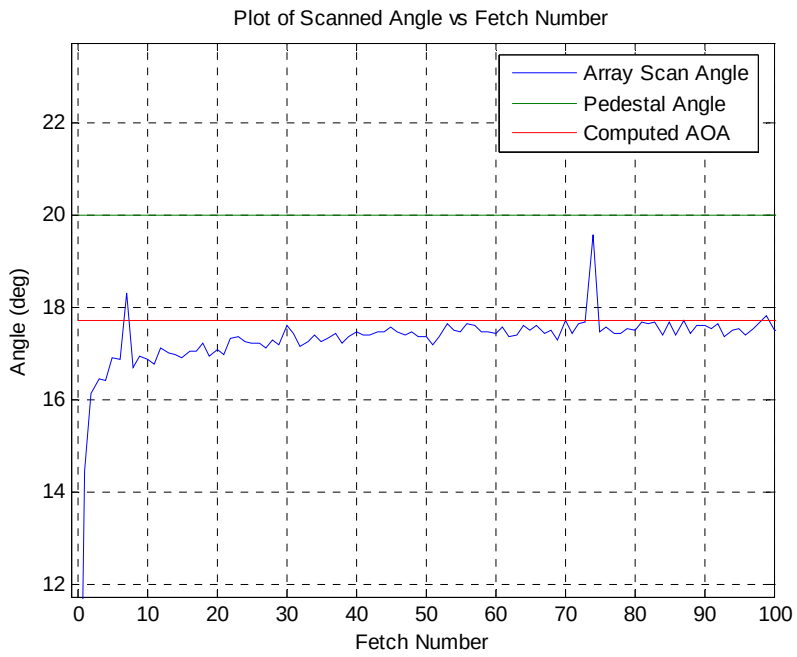



Figure 111. Plot of Correction Angle against Fetch Number for AOA of 20°

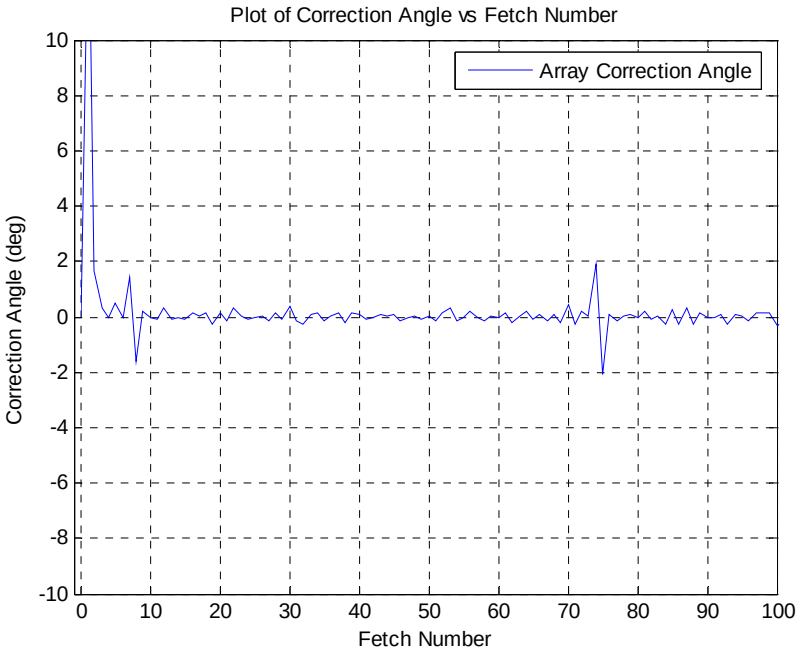



Figure 112. Plot of Correction Angle against Fetch Number for AOA of 20°

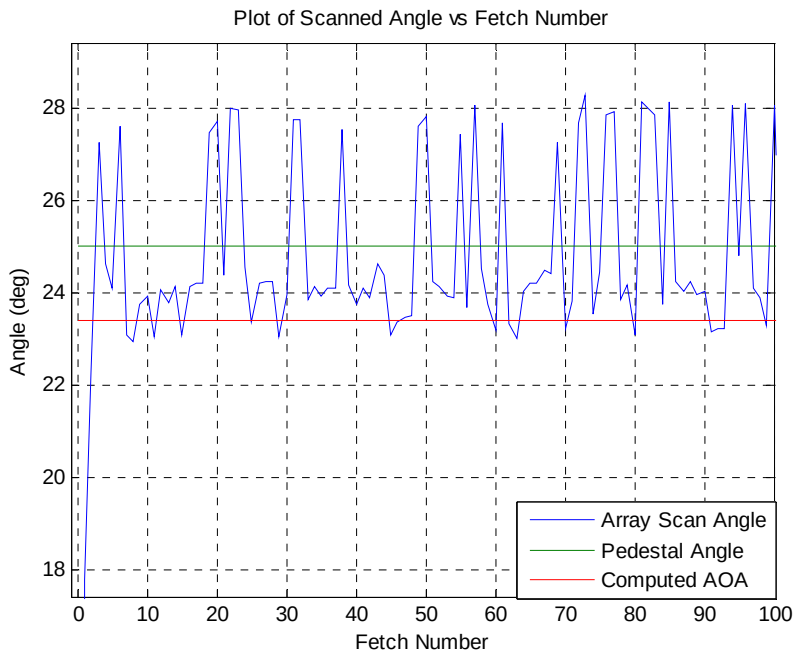



Figure 113. Plot of Angle against Fetch Number for AOA of 25°

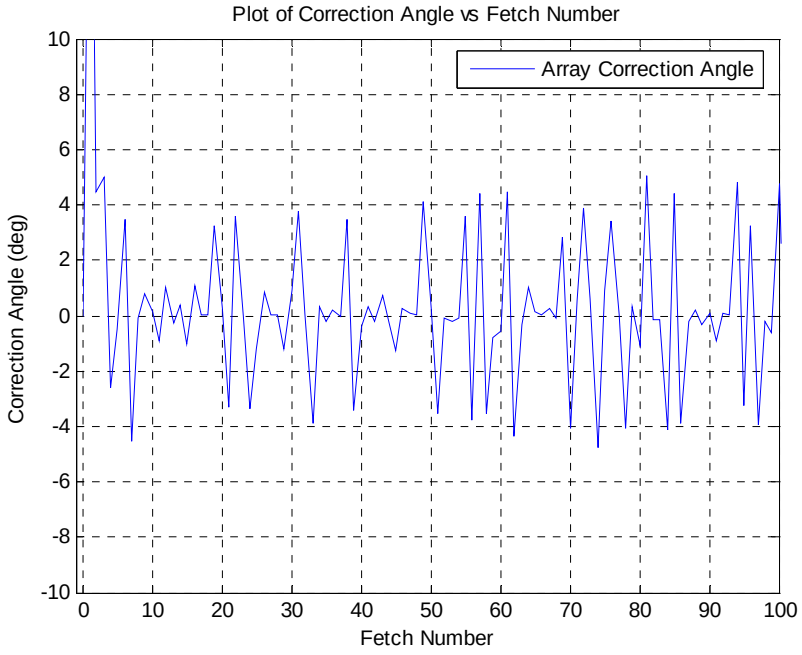



Figure 114. Plot of Correction Angle against Fetch Number for AOA of 25°


Figure 115. Plot of Correction Angle against Fetch Number for AOA of 30°

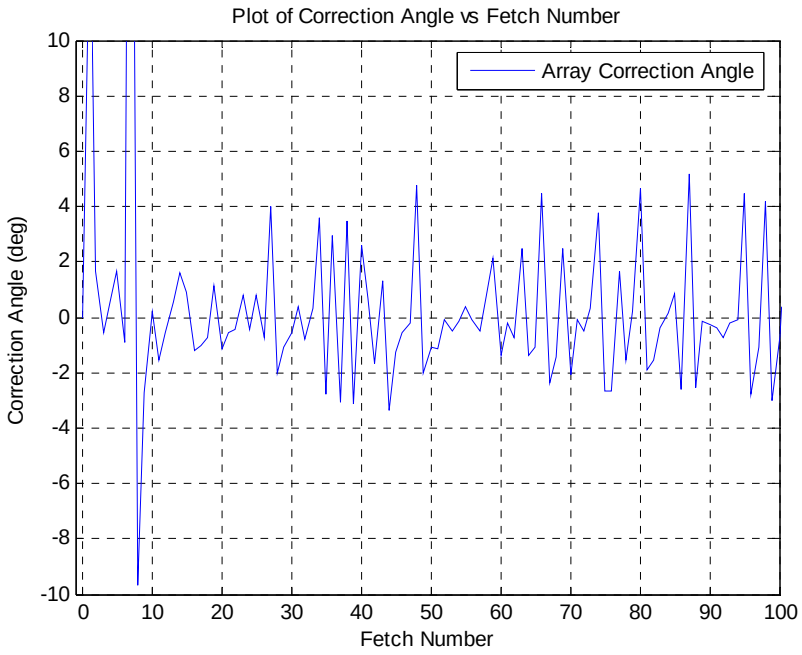



Figure 116. Plot of Correction Angle against Fetch Number for AOA of 30°

Several characteristics of the antenna array could be observed from the plots. First, there was a difference of approximately 3 degrees between the pedestal angle, which is the AOA of the wave from the transmitter, and the computed AOA obtained from the computation using the phase information obtained from each element. Most likely this is due to misalignment of the pedestal which was visually done. This could also be attributed in part to the antenna elements each having a different reference of $0^\circ$ despite the implementation of phase adjustments.

For each case, it was seen that the antenna array converges (within $1^\circ$) to a solution within 10 iterations of the module. This corresponding effect was observed in the plots of the correction angle. For each graph, the initial absolute value of the correction angle was very large as the antenna array attempted to make the correction from boresight to the AOA based on the angle computed using RSNS. Subsequently, the antenna array used monopulse DBF to converge to a solution. The correction angle was seen to approach to zero after several iterations of the module. This is consistent with what is expected using DBF.

An anomaly was noticed for the plots at $25^\circ$ and $30^\circ$. For those cases, the array antenna did not converge to a specific scan angle but instead computed a result that oscillated approximately $\pm 4^\circ$ of the expected scan angle. This could be attributed to reflections off of the side of the chamber, which is not symmetrical about the pedestal. This occurs only when the pedestal is rotated in the positive angle direction.

### 3. Tracking Test for Sweeping AOA at a Fixed Power Level

To determine the ability of the antenna array to acquire a UAV and effectively track it at range of AOAs, a test was set up to simulate a UAV moving across the antenna array, from $-30^\circ$ to $30^\circ$. To do this, the pedestal angle was rotated for the same angle range at $5^\circ$ increments and the output scan angle derived by the antenna array was obtained. Similar to the test in Section 2, the scan angle was plotted against fetch number. The transmitted power was set to a value such that the received power entering the

demodulator boards was approximately -56 dBm, The test was first conducted using RSNS to acquire and track the incoming wave and was performed twice to check the consistency of the results.

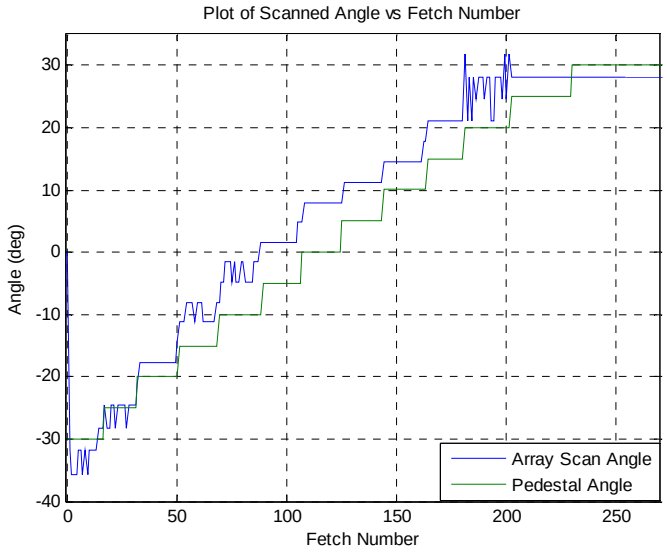



Figure 117. Plot of test 1 of angle against fetch number for AOA of $-30^{\circ}$ to $30^{\circ}$ using RSNS for tracking and acquisition

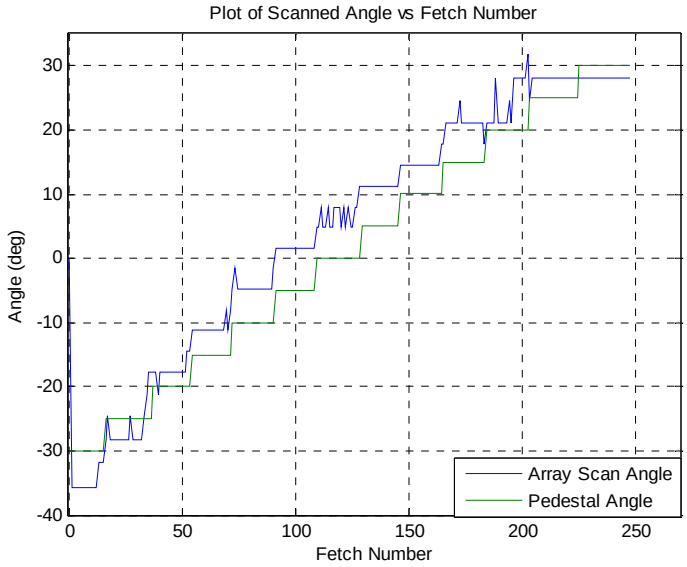



Figure 118. Plot of test 2 of angle against fetch number for AOA of $-30^{\circ}$ to $30^{\circ}$ using RSNS for tracking and acquisition

From the plots, it can be seen that using RSNS, the antenna array was able to follow the signal, achieving a 'step' output as the AOA was varied from $-30^{\circ}$ to $30^{\circ}$. Both test 1 and test 2 produced similar results. There was a constant difference between the 'steps' plotted for the pedestal angle and the output scan angle generated using RSNS. Since RSNS involves discrete bins for its output for a given input, there will be a observed difference between the pedestal angle and scan angle. Since the resolution provided by RSNS, using the [5, 7] moduli, is approximately $5^{\circ}$ at broadside, this difference is acceptable for the acquisition function.

At certain pedestal angles, it was noted that the output of the antenna array oscillated between two bins. This error was, however, not consistent between the two tests conducted. Since for this test, only the center two elements were used for RSNS computation, the received signal on each element could have varied slightly during the test for a particular AOA. This variation could have correspondingly caused the RSNS output bin selection to oscillate between two adjacent bins, which illustrates the grey code property of the RSNS.

The test was repeated using RSNS for acquisition and DBF for tracking. The sweep angles were the same as those used for the test for RSNS. The results are shown in Figure 119and Figure 120.

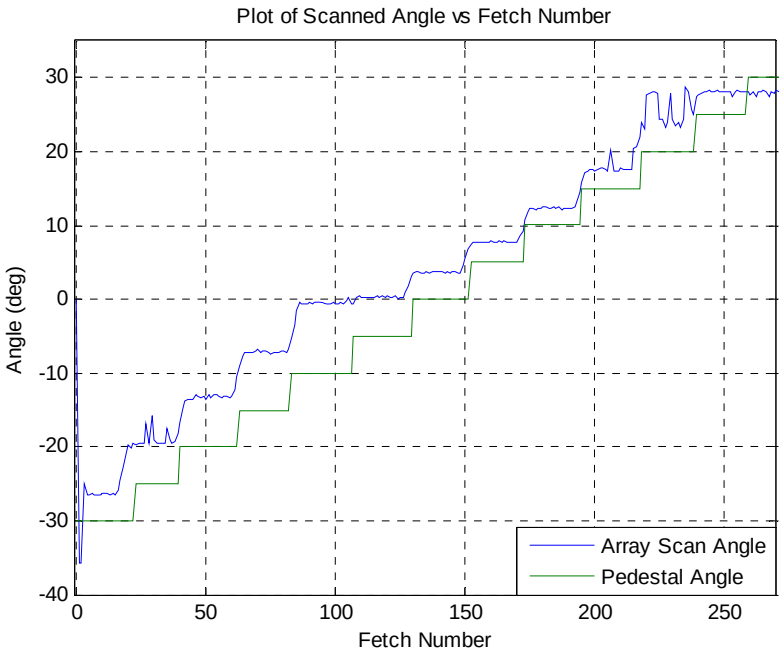



Figure 119. Plot of test 1 of angle against fetch number for AOA of $-30^{\circ}$ to $30^{\circ}$ using RSNS for acquisition and DBF for tracking

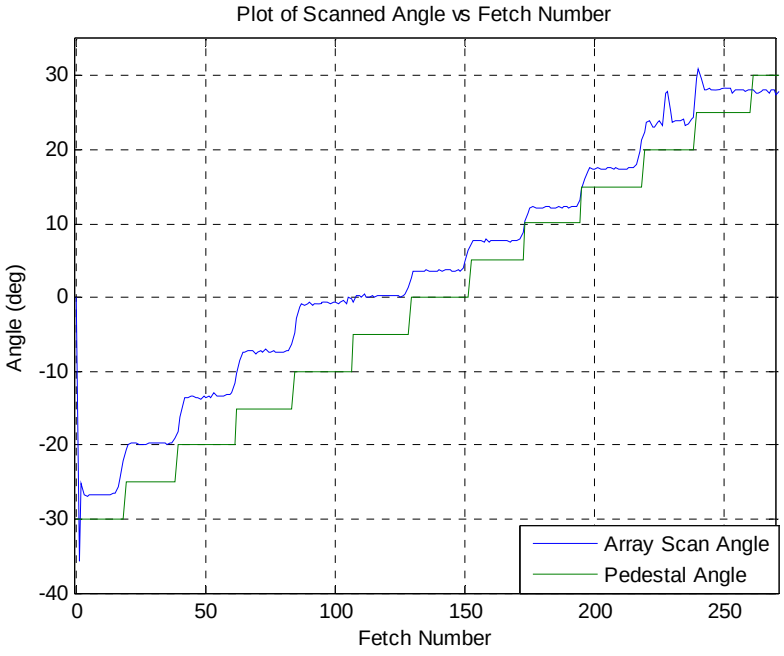



Figure 120. Plot of test 2 of angle against fetch number for AOA of $-30^{\circ}$ to $30^{\circ}$ using RSNS for acquisition and DBF for tracking

From the output generated for the RSNS-DBF option selected, it was observed that the antenna array was similarly able to acquire and track the incoming wave at different values of AOA. Similar to the tests conducted in Section 2, there was a noticeable difference of approximately $3^{\circ}$ degrees between the pedestal angle and the scan angle generated by the antenna array. The largest difference between the pedestal angle and the scan angle generated by the antenna array was at $-5^{\circ}$. At that angle, the antenna array generated a scan angle of approximately $0^{\circ}$. This is similar to the result obtained in Section 2 for the discrete case test at the same angle.

For the plots, it was observed that the antenna array was consistently able to converge to a particular AOA, with the exception when the AOA was $25^{\circ}$. In both test sequences, at that particular angle, an oscillation is observed for the antenna array output. This is consistent with what was obtained in Section 2 and is attributed to an inherent consequence of the setup within the chamber.

### 4. Tracking Test for Sweeping AOA at a Varying Power Levels

To understand the ability of the antenna array to track at different ranges, the transmitted power was varied to different values and the same tracking test conducted in Section 2 was repeated. The received power for each case was correspondingly noted. With a high received power, the situation where the UAV is close to the antenna array is simulated. In such a scenario, the high received power may potentially saturate the demodulator boards and cause an erroneous track. Conversely, with a low received power, there might be insufficient power for the demodulator boards to accurately determine the phase of the incoming signal, thus rendering it unable to track the target.

The test was first conducted using RSNS for direction finding. The transmitted power was varied such that the received power was at -11 dBm, -16 dBm, -26 dBm, -36 dBm, -46 dBm, -56 dBm, -63 dBm and -69 dBm. Figure 121 shows the graph of the antenna array using RSNS for the first four received power levels while Figure 122 shows the graph for the next four received power levels.

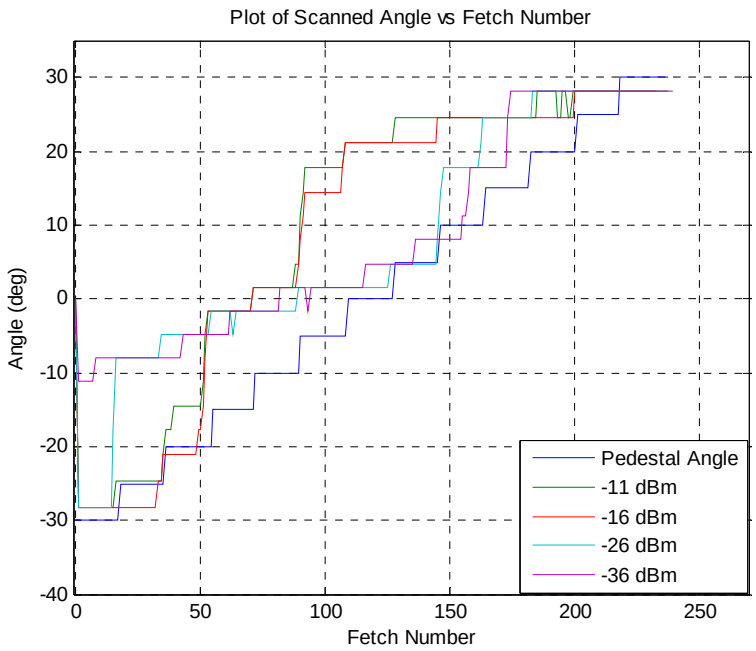



Figure 121. Plot of angle against fetch number for AOA of $-30^{\circ}$ to $30^{\circ}$ using RSNS received power levels of -11 dBm, -16 dBm, -26 dBm and -36 dBm

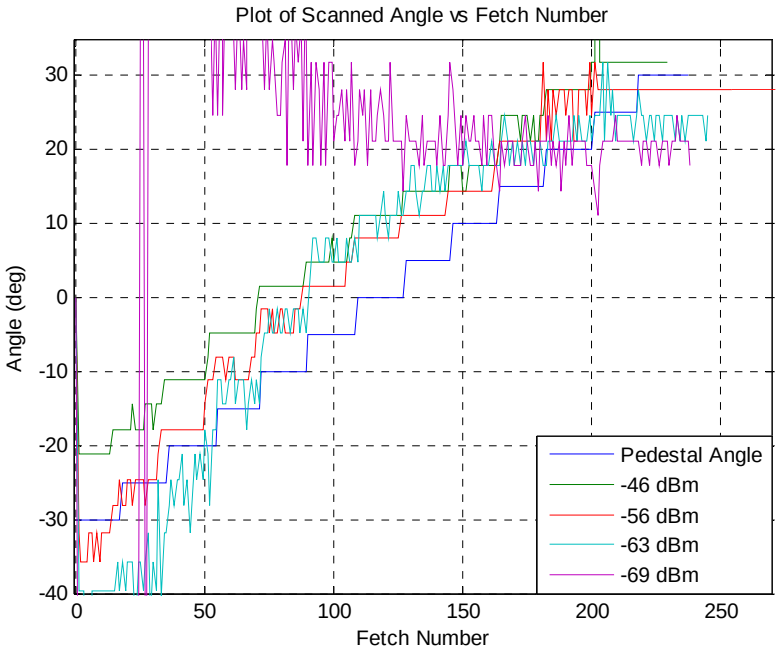



Figure 122. Plot of Angle against Fetch Number for AOA of $-30^{\circ}$ to $30^{\circ}$ using RSNS received power levels of -46 dBm, -56 dBm, -63 dBm and -69 dBm

At high received power levels as shown in Figure 121, it was observed that the step following fidelity decreased as the received power increased. For the received power of -26 dBm and -36 dBm, the antenna array output was able to track the direction of the incoming wave. The output of the array was similar to the step plot seen for the pedestal angle. When the power was increased above -26 dBm, the number of steps in the plots decreased. The antenna array, for power level of -11 dBm and -16 dBm, was not able to track every change in pedestal angle. This is seen clearly in the plot for the received power of -11 dBm. For that particular case, when the pedestal angle changed from $-20^{\circ}$ to $-10^{\circ}$ and from $5^{\circ}$ to $20^{\circ}$, the antenna array output remained constant, unable to track the change in AOA. This is likely due to saturation of the demodulator boards, which caused a loss in the resolution of the phase information in the $I$ and $Q$ components. This in turn caused the antenna array to not be able to distinguish small changes in AOA.

At low received power levels, shown in Figure 122, the antenna array behaves differently. As the power received is decreased below -46 dBm, antenna array output begins to oscillate about each of the steps. This is evident for received power levels of -56 dBm and -63 dBm, with the latter showing greater oscillation at each step. Although the output oscillates at these power levels, the antenna array is still able to track the different AOA. It is postulated that the oscillations are caused due to insufficient power received by the antenna elements, which causes a low SNR. When the power is decreased further to -69 dBm, the array is no longer able to track the incoming signal. The effect was predicted by Benveniste's simulations [10].

The same test was then repeated using RSNS for acquisition and monopulse DBF for tracking. The received power was set at successive discrete values of -11 dBm, -16 dBm, -26 dBm, -36 dBm, -46 dBm, -56 dBm, -61 dBm and -63 dBm. Figure 123 and Figure 124 show the graphs of the antenna array using RSNS/DBF at the various power levels.

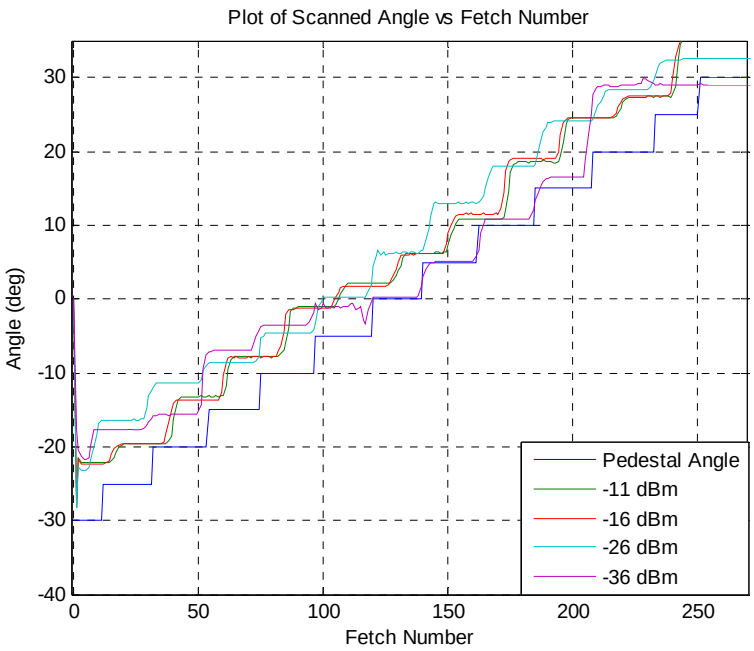



Figure 123. Plot of Angle against Fetch Number for AOA of $-30^{\circ}$ to $30^{\circ}$ using RSNS/DBF for received power levels of -11 dBm, -16 dBm, -26 dBm and -36 dBm

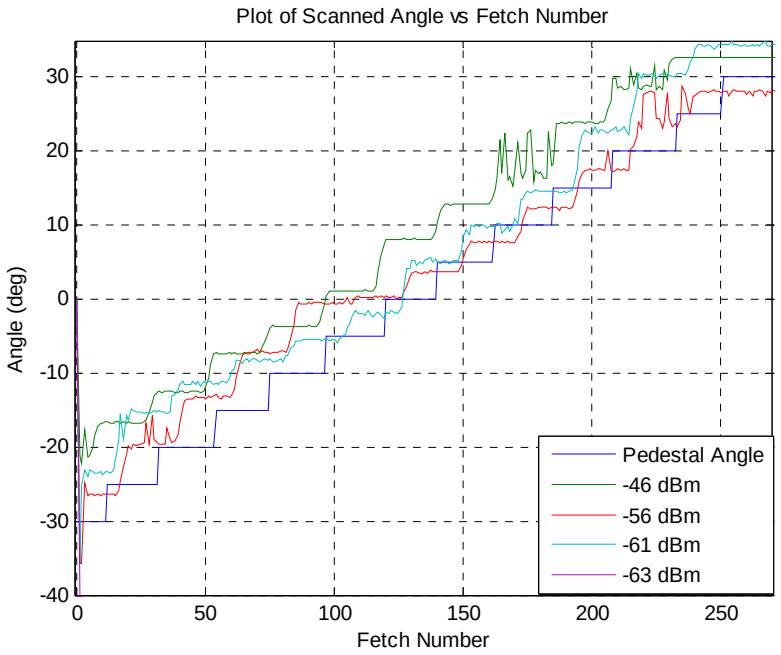



Figure 124. Plot of Angle against Fetch Number for AOA of $-30^{\circ}$ to $30^{\circ}$ using RSNS/DBF for received power levels of -46 dBm, -56 dBm, -61 dBm and -63 dBm

From Figure 123, it was observed that even as the received power increased to -11 dBm, the antenna array was able to track the AOA of the incoming signal. When the same test was conducted using RSNS, at -11 dB, the antenna array was no longer able to accurately track the incident wave. The difference between the two methods of tracking could be due to the sensitivity of the element's phase information. For RSNS, only elements 3 and 4 were used to obtain the AOA whereas, for DBF, all six elements are used to derive the AOA. Using additional elements in computing the AOA may prove advantageous when there is a possibility of element saturation.

When the received power was reduced, the performance of the antenna array remained relatively unchanged up to -61 dBm. At lower received power levels, it is observed that the antenna array takes a longer time for the output scan angle to converge. At -61 dBm, the antenna array does not converge to a scan angle when the AOA is changed but instead continuously oscillates about the computed angle. As explained in the RSNS test, the oscillations are due to low SNR. When the received power was decreased to -69 dB, the antenna array was no longer able to track the signal.

## D. SUMMARY OF RESULTS

In this chapter, several key results were obtained. For the software modules, it was shown that the *NTSC Decoding Module* developed was capable of decoding NTSC encoded video signal in monochrome. The test was conducted for a single channel, achieving a frame rate of approximately one frame per second.

The *FM Demodulation Module* made some headway in demodulating FM encoded signals received on a single channel. The module, however, was only successful in decoding sinusoidal input waveforms and was not able to do the same for square and triangle shaped input waveforms.

The *Tracking Array Module*, when tested during the bench top test with the phase shifters, was able to track the AOA for a waveform that arrived from $-10^{\circ}$, $0^{\circ}$ and $10^{\circ}$. Due to hardware limitation within the lab, the bench top test was conducted for only three discrete cases. This was sufficient to verify the functionality of the module prior to the test in the anechoic chamber.

When the test was conducted in the anechoic chamber, it was seen that the antenna array was able to track varying AOA from $-30^{\circ}$ to $30^{\circ}$ using both RSNS and monopulse DBF for direction finding. When power levels were reduced, both RSNS and DBF direction finding behaved similarly in terms of an oscillating output. Both methods of direction finding were still able to track the incoming signal but did not converge to a solution at each step. At higher power levels, direction finding using the monopulse DBF was observed to be more accurate than RSNS in attempting to correctly distinguish the change in AOA at each step. RSNS was still able to track the general change in AOA but was unable to maintain the resolution of following every change in AOA. This was attributed to element saturation. The effect of saturation did not affect DBF in a large manner as the antenna array was still capable of tracking the incoming signal at high received power levels.

# VII. CONCLUSIONS AND RECOMMENDATIONS

## A. SUMMARY AND CONCLUSIONS

The objective of the research presented in this thesis, as part of an ongoing project, was to develop a digital tracking array for use in communications with a UAV using commercial-off-the-shelf (COTS) components. Research had previously been done in key areas using RSNS and DBF for direction finding and this research sought to extend the work, as well as integrating it for a complete solution. Additional steps were taken to extend the communications aspect of the project with focus on modulation techniques and video decoding for eventual future integration with the developed tracking antenna.

The focus on the hardware portion was to integrate all hardware modules developed for the complete system to work at optimum performance levels. Calculation for the system power budget was done to work out the required power levels. This required the careful calibration of the power levels provided across all modules, taking into account the optimum levels required by the inputs such as the demodulator LO drive power level. Care was also taken in the integration of the hardware modules to high levels of workmanship to ensure that no failure should be attributed to workmanship problems such as a loose cable connection or insufficient power level during testing. This resulted in time savings as failures or inconsistencies in the test results can be quickly nailed down rather than spending additional effort to include the hardware during troubleshooting. Measurements were also done for the new two-element subarray antenna to check the measured performance against expected simulation results. Based on data obtained from antenna measurements and power budget calculations, calculation of the performance envelope for the system was presented.

The software for the antenna array was developed using a modular approach to facilitate testing and development, and to create avenues for future extension. Three separate modules were developed: NTSC Decoding, FM Demodulation and Monopulse Beamforming and Tracking. The NTSC Decoding module was successfully implemented

for a single channel array, with the module able to decode incoming video signals and display the generated output video in monochrome on a terminal. The FM Demodulation Module was developed and tested successfully with sinusoidal input waveform for a single channel array. The Beamforming and Tracking Module used past research work done on RSNS and combined it with additional research conducted in monopulse DBF to form a module capable of using both these techniques to acquire and track a UAV using the six-element antenna array.

The hardware and software solutions were integrated and tested using a bench top test setup and subsequently a test was conducted in an anechoic chamber to characterize the antenna array. The antenna array was capable of acquiring and tracking the AOA of the incident waveform as the AOA was varied from $-30^{\circ}$ to $30^{\circ}$. The performance of the antenna array was tested at different levels of received power to investigate the performance capability of the system.

## B. RECOMMENDATIONS FOR FUTURE WORK

Based on the work done for this research, there are several avenues of research and development that can be explored to extend the capabilities of this project. These thrusts represent the current direction for the antenna array, of developing an array antenna based on FM demodulation to receive NTSC encoded video data from a UAV. With the acquisition and tracking capability of the array completed, emphasis can now be placed on development of the communication aspect of the array.

### 1. Extension of FM Demodulation Module for NTSC Video Signal

The FM Demodulation Module developed in this thesis was a test setup using primarily sinusoidal input waveform. If FM remains the selected modulation technique for communications with the UAV, the module must be extended to manage different types of waveforms, such as square- and saw tooth-shaped functions. Eventually, this module will have to be extended from a single channel module to a six channel module for it to be integrated with current array hardware.

### 2. Extension of NTSC Decoding Module to Six Channel

The NTSC Decoding Module was capable of receiving and decoding video data received on a single channel. Future work should look into extending this to a module capable of receiving video data from six channels. Additional work could also be done to modify the current module to receive and decode color information.

### 3. Increasing Bin Resolution for RSNS Direction Finding

The RSNS module used in this thesis used [5, 9] moduli that resulted in bin size of approximately $5^0$ at boresight. By increasing the values used for the moduli, the bin size will decrease, thereby allowing the antenna array to be able to track the incoming signal with higher accuracy.

### 4. Variable Slope Constant

In the tracking array monopulse beamforming, a single value for the slope constant was used. To increase the speed of convergence, a variable value of the slope constant could be used based on the current scan angle.

### 5. Quadrature Demodulator

Low second-order input intercept point (IIP2) specifications lead to higher second-order distortion, which limits the performance of the quadrature demodulator. Newer models are now available that have significantly higher IIP2 specifications. One such model is LT5575 from Linear Technologies. This IC has an IIP2 specification in excess of +50 dBm. This is almost twice the specification of +25.5 dBm for the current AD8347 demodulator.

### 6. ADC and Modulation Scheme

Several specifications of the ADC are key to further the performance of the receiver. A higher resolution ADC has lower noise, hence better sensitivity. Also, a higher resolution ADC allows for higher modulation schemes. The current modulation

scheme (FM) is a well established analog modulation scheme that is used in many communication systems. However, it also forms the backbone for other digital communication schemes, such as Frequency Shift Keying (FSK) and other multi-level signaling modulation schemes [39]. Therefore, adoption of multi-level signaling modulation will allow the amount of data that can be transmitted in a given bandwidth to increase.

In addition, the use of an ADC with a much higher sampling rate, of at least 5 Gs/s, would eliminate the need for a separate hardware demodulator. A sampling rate of 5 Gs/s ensures that signals acquired at 2.4 GHz will not be aliased. With the demodulation and processing done in the digital domain, the receiver system has the potential to be reconfigurable quickly to receive other types of signals and at other frequencies.

### 7. LO

The VCO used to provide the LO signal can be changed to a digital controllable oscillator such as the Lab Brick LSG-402 Signal Generator so that the frequency of the LO signal can be adjusted digitally to match exactly the frequency of the received RF signal for homodyning.

### 8. System Enhancements

The current system architecture allows for upgrade to a multi-carrier system as a single hardware solution implemented on the direct-conversion architecture offers more freedom in addressing multiple bands of operation. As there is only a single set of hardware for multiple bands, this solution promises to be cost effective. At the same time, if the recommendations above are implemented, the system can be combined with the work currently undertaken by Yeo on High Resolution Spectrum Estimation for Digital Tracking Array [40] to detect the carrier frequency of signals from desired UAVs and use it to adjust the frequency of the LO to enable homodyning. This scenario affords the user freedom to receive multiple signals at different frequencies from different UAVs simultaneously. A transmit module can be added to each channel to convert the receiver system to one that can transmit and receive (T/R). This option upgrades the system to possess the capability to communicate with the UAV, or turn it into phased array radar.

# APPENDIX

## A. MATLAB CODE FOR DBF SIMULATION PROGRAM

```
% Sum_Beam_6elm_menu.m

%*********************************************************************
% This m file computes the correction angle for a system using DBF
% for a given angle of incidence. The file is able to also work with
% a fixed phase antenna element output. The second half of the program
% plots the sum and difference beams for a given scan angle. This is
% used to compute the monopulse slope constant at different scan angles.
%
%written by MAJ Devieash James Pandya, Singapore, 2009.
%*********************************************************************
clc;
clear;

N = 6;
X = [0, 0; 0, 0];

theta_s = -30;
theta = -27;

slope_constant = -0.076;

rad=pi/180;
f=2.4e09;
lambda=3e08/f;
d=0.065;

theta = input('\n Enter in angle of arrival ');
theta_s = input('\n Enter in start scan angle ');
k=2*pi/lambda;

for i=1:6
        element_d(i) = (2*i-(N+1))/2 * d;
end

loop = 0;

while loop == 0
    I=[real(exp(j*k*element_d(1)*sind(theta)))
real(exp(j*k*element_d(2)*sind(theta)))
real(exp(j*k*element_d(3)*sind(theta)))
real(exp(j*k*element_d(4)*sind(theta))) ...
      real(exp(j*k*element_d(5)*sind(theta)))
real(exp(j*k*element_d(6)*sind(theta)))];
```

```
    Q=[imag(exp(j*k*element_d(1)*sind(theta)))
imag(exp(j*k*element_d(2)*sind(theta)))
imag(exp(j*k*element_d(3)*sind(theta)))
imag(exp(j*k*element_d(4)*sind(theta))) ...
     imag(exp(j*k*element_d(5)*sind(theta)))
imag(exp(j*k*element_d(6)*sind(theta)))];

    %I=[cos(j*k*element_d(1)/2*sind(theta))
cos(j*k*element_d(2)/2*sind(theta)) cos(j*k*element_d(3)/2*sind(theta))
cos(j*k*element_d(4)/2*sind(theta))]
    %Q=[sin(j*k*element_d(1)/2*sind(theta))
sin(j*k*element_d(2)/2*sind(theta)) sin(j*k*element_d(3)/2*sind(theta))
sin(j*k*element_d(4)/2*sind(theta))]


    A=sqrt(I.^2+Q.^2); %Magnitude
    P=atan2(Q,I)*180/pi %Phase


    %prog_data = [-118.8, -127.46, -114.75, -116.28, 170.78, -166.04];
    prog_data = [-289.9 -289 -55 1.74 -115 -98];

    prog_data = rad * prog_data;
    weights = 0;
    test_values = 0;

    for counter = 1:6
      weights(counter) = exp(-j*k*element_d(counter)*sin(theta_s*rad));
     % test_values(counter) =
weights(counter)*exp(j*atan2(Q(counter),I(counter)));
     test_values(counter) = weights(counter)*exp(j*prog_data(counter));

    end

    %weights(1) = exp(-j*k*element_d(1)*sin(theta_s*rad));
    %weights(2) = exp(-j*k*element_d(2)*sin(theta_s*rad));
    %weights(3) = exp(-j*k*element_d(3)*sin(theta_s*rad));
    %weights(4) = exp(-j*k*element_d(4)*sin(theta_s*rad));


    %test_1 = weights(1)*exp(j*atan2(Q(1),I(1)))
    %test_2 = weights(2)*exp(j*atan2(Q(2),I(2)))
    %test_3 = weights(3)*exp(j*atan2(Q(3),I(3)))
    %test_4 = weights(4)*exp(j*atan2(Q(4),I(4)))


    Sum_beam_plot =
test_values(1)+test_values(2)+test_values(3)+test_values(4)+test_values(
5)+test_values(6)
    Diff_beam_plot = test_values(1)+test_values(2)+test_values(3)-
test_values(4)-test_values(5)-test_values(6)
```

```
    Correction_Angle = imag(Diff_beam_plot / Sum_beam_plot)
/slope_constant;
    theta_s = theta_s + Correction_Angle;
    msg = sprintf('\nCorrection angle is %f. \nNew scan angle is
%f',Correction_Angle, theta_s);
    disp (msg)

    if abs (Correction_Angle) < 1E-3
        break;
    end

    loop = input('\n To continue tracking, press 0, else press 1 \n');
end

if input('To exit program, press 1, else, press 0 to proceed with
pattern plots ') == 1
    break;
end

theta_s = input('\n Enter in scan angle for the pattern plots ');

for counter = 1:6
      weights(counter) = exp(-j*k*element_d(counter)*sin(theta_s*rad));
end

index = 1;

for counter = -90:0.1:90

    theta = counter;

    I=[real(exp(j*k*element_d(1)*sind(theta)))
real(exp(j*k*element_d(2)*sind(theta)))
real(exp(j*k*element_d(3)*sind(theta)))
real(exp(j*k*element_d(4)*sind(theta))) ...
     real(exp(j*k*element_d(5)*sind(theta)))
real(exp(j*k*element_d(6)*sind(theta)))];
    Q=[imag(exp(j*k*element_d(1)*sind(theta)))
imag(exp(j*k*element_d(2)*sind(theta)))
imag(exp(j*k*element_d(3)*sind(theta)))
imag(exp(j*k*element_d(4)*sind(theta))) ...
     imag(exp(j*k*element_d(5)*sind(theta)))
imag(exp(j*k*element_d(6)*sind(theta)))];

    Sum_beam_plot(index) = weights(1)*exp(j*atan2(Q(1),I(1))) +
weights(2)*exp(j*atan2(Q(2),I(2)))+ weights(3)*exp(j*atan2(Q(3),I(3))) +
weights(4)*exp(j*atan2(Q(4),I(4))) + ...
        weights(5)*exp(j*atan2(Q(5),I(5))) +
weights(6)*exp(j*atan2(Q(6),I(6)));
```

```
    Diff_beam_plot(index) = weights(1)*exp(j*atan2(Q(1),I(1))) +
weights(2)*exp(j*atan2(Q(2),I(2)))- weights(3)*exp(j*atan2(Q(3),I(3))) -
weights(4)*exp(j*atan2(Q(4),I(4))) - ...
        weights(5)*exp(j*atan2(Q(5),I(5))) -
weights(6)*exp(j*atan2(Q(6),I(6)));

  %  Sum_beam_plot(index) = weights(1)*exp(j*prog_data(1)) +
weights(2)*exp(j*prog_data(2))+ weights(3)*exp(j*prog_data(3)) +
weights(4)*exp(j*prog_data(4)) + ...
  %      weights(5)*exp(j*prog_data(5)) +
weights(6)*exp(j*prog_data(6));
  %  Diff_beam_plot(index) = weights(1)*exp(j*prog_data(1)) +
weights(2)*exp(j*prog_data(2))- weights(3)*exp(j*prog_data(3)) -
weights(4)*exp(j*prog_data(4)) - ...
  %      weights(5)*exp(j*prog_data(5)) -
weights(6)*exp(j*prog_data(6));

    Sum_Diff(index) = imag( Diff_beam_plot(index)/Sum_beam_plot(index));

    index = index + 1;
end
i = [-90:0.1:90];
figure(1)

plot (i, Sum_beam_plot,i, imag(Diff_beam_plot) )
grid on;

figure(2)
plot(i,20*log10(Sum_beam_plot))
grid on
figure(3)
plot(i, Sum_Diff);
grid on

figure(4)
plot(i,20*log10(imag(Diff_beam_plot)))
grid on
axis([-90, 90, -60, 10])


Pr=polyfit(i,Sum_Diff,1);
R2=polyval(Pr,1)


%figure(2)
%plot (array, 20*log10(abs(Sum_beam_plot)/max(Sum_beam_plot)), array,
20*log10(abs(Diff_beam_plot)/max(Sum_beam_plot)), array,
20*log10(Sum_Diff))
grid on;
```

## B. MATLAB CODE FOR CALCULATION OF THE GAIN FOR THE TWO-ELEMENT SUBARRAY ANTENNA

```
% thesis_antenna.m
% This code calculates and plots the gain pattern of the two-element
% subarray with measured data.

close all
clear
clc

load Narda645_h.txt
load Narda645_v.txt
load DDipole_v.txt
load DDipole_h.txt

G_narda645 = 17.12;% Gain value obtained from figure Narda645 at
f=2.4GHz
maxSL_ref_h = 0.36801;% Max recorded value obtained from Narda645_h
maxSL_ref_v = -1.4651;% Max recorded value obtained from Narda645_v

DDipole_v(:,3) = G_narda645-maxSL_ref_v+DDipole_v(:,2);
DDipole_h(:,3) = G_narda645-maxSL_ref_h+DDipole_h(:,2);

figure(1)
%plot (DDipole_v(:,1),DDipole_v(:,2))
%hold on
plot (DDipole_v(:,1),DDipole_v(:,3))
grid on
title ('Gain Pattern of Two Element Sub-array - Azimuth')
xlabel ('degrees')
ylabel ('Gain (dB)')
figure (2)
polardB (DDipole_v(:,1),DDipole_v(:,3), 'r')
title ('Gain Pattern of Two Element Sub-array - Azimuth')

figure (3)
plot (DDipole_h(:,1),DDipole_h(:,3))
grid on
title ('Gain Pattern of Two Element Sub-array - Elevation')
xlabel ('degrees')
ylabel ('Gain (dB)')
figure (4)
polardB (DDipole_h(:,1),DDipole_h(:,3), 'r')
title ('Gain Pattern of Two Element Sub-array - Elevation')
```

## C. MATLAB CODE FOR SYSTEM PERFORMANCE ANALYSIS

```
% thesis_system_cal2.m
% This code calculates and plots the performance envelope of the
% receiver array for analysis.

clear
close all
clc

h = 6.626*10e-34; %planck's constant
k = 1.38e-23;
T = 290;
c = 3e8;
f = 2.4e9;
lambda = c/f;
B = 9e7; % BW of demod

N = k*T*B;
N_dBm = 10*log10(k*T*B)+30;

%NF_dB = 1.8; % system NF at max gain
%NF_dB = 37; % system NF at min gain

NF_lna_dB = 1.7;
NF_demod_dB = 11.8;
G_lna_dB = 25;
G_lna = 10^(G_lna_dB/10);

T_lna = 290*((10^(NF_lna_dB/10))-1);
T_demod = 290*((10^(NF_demod_dB/10))-1);
T_A = h*f/k*(1+1/(exp(h*f/(k*T))-1))+7*10^26/f^3+2.726+f^2.5/10^25;
T_sys = T_A+T_lna+T_demod/G_lna;

Pmin_demod_dBm = -65; % from start of linear region of TPC30 on
datasheet
Pmin_demod = 10^((Pmin_demod_dBm-30)/10) %W

%No_elems = 1;
No_elems = 6;

Pt = 100e-3; % Power transmitted
Gt = 10^0.3; % gain of transmitter on UAV
Gr = 10^1; % gain of single sub-array element = 2*1.64*2 (double dipole
over ground plane)
uav_altitude = 3000 * 0.3048; % 3000 ft

Pmin_ch_w_lna_dBm = Pmin_demod_dBm-G_lna_dB % after antenna
Pmin_ch_w_lna = 10^((Pmin_ch_w_lna_dBm-30)/10) %W

SNR_dB = 12; % min for good demod of FM
SNR_sys_dB = Pmin_ch_w_lna_dBm-N_dBm+10*log10(No_elems)
```

```
Range_w_lna = sqrt((Pt*Gt*Gr*6*lambda^2/((4*pi)^2*Pmin_ch_w_lna))-
uav_altitude^2)
Range_wo_lna = sqrt((Pt*Gt*Gr*6*lambda^2/((4*pi)^2*Pmin_demod))-
uav_altitude^2)

%-----------------------------------
earth_radius = 6378e3;
earth_model = 4/3;
earth_radius_adj = earth_radius * earth_model;

radar_horizon = sqrt(2* earth_radius_adj * uav_altitude);

HPBW = 31;

%-----------------------------------
% theta and phi orientation switched....
load 3Dpattern.txt

dir = X3Dpattern(:,3);
Gain_3dpattern1_dB = reshape(dir,37,[]); % column - theta values fr 0 -
180 deg for each phi value
Gain_3dpattern1_dB = Gain_3dpattern1_dB(1:37,1:37); % only phi = 0-180
deg
Gain_3dpattern1 = 10.^(Gain_3dpattern1_dB./10);

phi1 = 0:5:180;
theta1 = 0:5:180;

figure (6);
meshc(phi1, theta1, Gain_3dpattern1_dB)
ylabel ('\theta');
xlabel ('\phi');
zlabel ('Directivity (dB)');
title ('Simulated 3D Gain Pattern of Sub-Array Beam')

Gain_elevation_dB(1,:) = Gain_3dpattern1_dB(19,:); % elevation gain
(theta = 90 deg cut)
Gain_elevation(1,:) = Gain_3dpattern1(19,:);
Gain_elevation(2,:) = -75:5:105; % tilt antenna 15 deg upwards

figure (7);
polardB (Gain_elevation(2,:), Gain_elevation_dB(1,:), 'r')
% plot (Gain_elevation(2,:), Gain_elevation_dB(1,:))
% ylabel ('Directivity (dB)');
% xlabel ('\phi');
title ('Simulated Elevation Gain Pattern of Sub-Array Beam with 15^o
tilt')

%-----------------------------------
```

```
k = 1;

for i = 1:1:37
    Gain_new = 6*Gain_elevation(1,i);

    range_angle_matrix(k,1) = i*5-95;
    range_angle_matrix_LNA(k,1) = i*5-95;

    Pt = 100e-3;
    range_angle_matrix(k,2) = sqrt((Pt*Gt*Gain_new*lambda^2/((4*pi)^2 *
Pmin_demod))-uav_altitude^2)/1000;
    range_angle_matrix_LNA(k,2)=
sqrt((Pt*Gt*Gain_new*G_lna*lambda^2/((4*pi)^2 * Pmin_demod))-
uav_altitude^2)/1000;

    Pt = 200e-3;
    range_angle_matrix(k,3) = sqrt((Pt*Gt*Gain_new*lambda^2/((4*pi)^2 *
Pmin_demod))-uav_altitude^2)/1000;
    range_angle_matrix_LNA(k,3)=
sqrt((Pt*Gt*Gain_new*G_lna*lambda^2/((4*pi)^2 * Pmin_demod))-
uav_altitude^2)/1000;

    Pt = 500e-3;
    range_angle_matrix(k,4) = sqrt((Pt*Gt*Gain_new*lambda^2/((4*pi)^2 *
Pmin_demod))-uav_altitude^2)/1000;
    range_angle_matrix_LNA(k,4)=
sqrt((Pt*Gt*Gain_new*G_lna*lambda^2/((4*pi)^2 * Pmin_demod))-
uav_altitude^2)/1000;

    Pt = 1;
    range_angle_matrix(k,5) = sqrt((Pt*Gt*Gain_new*lambda^2/((4*pi)^2 *
Pmin_demod))-uav_altitude^2)/1000;
    range_angle_matrix_LNA(k,5)=
sqrt((Pt*Gt*Gain_new*G_lna*lambda^2/((4*pi)^2 * Pmin_demod))-
uav_altitude^2)/1000;

    Pt = 1.5;
    range_angle_matrix(k,6) = sqrt((Pt*Gt*Gain_new*lambda^2/((4*pi)^2 *
Pmin_demod))-uav_altitude^2)/1000;
    range_angle_matrix_LNA(k,6)=
sqrt((Pt*Gt*Gain_new*G_lna*lambda^2/((4*pi)^2 * Pmin_demod))-
uav_altitude^2)/1000;

    Pt = 2;
    range_angle_matrix(k,7) = sqrt((Pt*Gt*Gain_new*lambda^2/((4*pi)^2 *
Pmin_demod))-uav_altitude^2)/1000;
    range_angle_matrix_LNA(k,7)=
sqrt((Pt*Gt*Gain_new*G_lna*lambda^2/((4*pi)^2 * Pmin_demod))-
uav_altitude^2)/1000;

    k = k+1;
end
```

```
figure(1);
plot (range_angle_matrix(:,1), range_angle_matrix(:,2),
range_angle_matrix(:,1), range_angle_matrix(:,3),
range_angle_matrix(:,1), range_angle_matrix(:,4), ...
    range_angle_matrix(:,1), range_angle_matrix(:,5),
range_angle_matrix(:,1), range_angle_matrix(:,6),
range_angle_matrix(:,1), range_angle_matrix(:,7));
ylabel ('Ground Range (km)');
xlabel ('Angle \theta');
grid on;
Legend ('P_t = 100 mW', 'P_t = 200 mW','P_t = 500 mW', 'P_t = 1 W', 'P_t
= 1.5 W', 'P_t = 2 W');

title('Plot of Range vs Angle (Without LNA)')

figure(2);
plot (range_angle_matrix_LNA(:,1), range_angle_matrix_LNA(:,2),
range_angle_matrix_LNA(:,1), range_angle_matrix_LNA(:,3),
range_angle_matrix_LNA(:,1), range_angle_matrix_LNA(:,4), ...
    range_angle_matrix_LNA(:,1), range_angle_matrix_LNA(:,5),
range_angle_matrix_LNA(:,1), range_angle_matrix_LNA(:,6),
range_angle_matrix_LNA(:,1), range_angle_matrix_LNA(:,7));

ylabel ('Ground Range (km)');
xlabel ('Angle \theta');
grid on;
Legend ('P_t = 100 mW', 'P_t = 200 mW','P_t = 500 mW', 'P_t = 1 W', 'P_t
= 1.5 W', 'P_t = 2 W');
title('Plot of Range vs Angle (With LNA)')
grid on;

% Computation of received Power for different ranges when UAV is flying
% at 3000 feet

theta = -15:5:70;

r = (uav_altitude./tan((theta+15).*(pi/180)))/1000;
range_target = sqrt ((r*1000).^2 + uav_altitude.^2);

Gain_new = Gain_3dpattern1(19,17:34); % elevation gain (theta = 90 deg
cut)

Pt = 100e-3;
range_power_rcvd_100mw =
Pt*Gt*Gain_new*lambda^2./((4*pi)^2*range_target.^2)*1e9;
range_power_rcvd_LNA_100mw=
Pt*Gt*Gain_new*G_lna*lambda^2./((4*pi)^2*range_target.^2)*1e6;

Pt = 200e-3;
range_power_rcvd_200mw =
Pt*Gt*Gain_new*lambda^2./((4*pi)^2*range_target.^2)*1e9;
```

```
range_power_rcvd_LNA_200mw =
Pt*Gt*Gain_new*G_lna*lambda^2./((4*pi)^2*range_target.^2)*1e6;

Pt = 500e-3;
range_power_rcvd_500mw =
Pt*Gt*Gain_new*lambda^2./((4*pi)^2*range_target.^2)*1e9;
range_power_rcvd_LNA_500mw =
Pt*Gt*Gain_new*G_lna*lambda^2./((4*pi)^2*range_target.^2)*1e6;

Pt = 1;
range_power_rcvd_1w =
Pt*Gt*Gain_new*lambda^2./((4*pi)^2*range_target.^2)*1e9;
range_power_rcvd_LNA_1w =
Pt*Gt*Gain_new*G_lna*lambda^2./((4*pi)^2*range_target.^2)*1e6;

Pt = 1.5;
range_power_rcvd_1_5w =
Pt*Gt*Gain_new*lambda^2./((4*pi)^2*range_target.^2)*1e9;
range_power_rcvd_LNA_1_5w =
Pt*Gt*Gain_new*G_lna*lambda^2./((4*pi)^2*range_target.^2)*1e6;

Pt = 2;
range_power_rcvd_2w = Pt*Gt*Gain_new*lambda^2./((4*pi)^2 *
range_target.^2)*1e9;
range_power_rcvd_LNA_2w=
Pt*Gt*Gain_new*G_lna*lambda^2./((4*pi)^2*range_target.^2)*1e6;

figure(3);
plot (r, range_power_rcvd_100mw, r, range_power_rcvd_200mw, r,
range_power_rcvd_500mw, ...
    r, range_power_rcvd_1w, r, range_power_rcvd_1_5w, r,
range_power_rcvd_2w);
ylabel ('Power Received (nW)');
xlabel ('Ground Range to UAV (km)');
grid on;
Legend ('P_t = 100 mW', 'P_t = 200 mW','P_t = 500 mW', 'P_t = 1 W', 'P_t
= 1.5 W', 'P_t = 2 W');

title('Plot of Received Power vs Ground Range(Without LNA)')

figure(4);
plot (r, range_power_rcvd_LNA_100mw, r, range_power_rcvd_LNA_200mw, r,
range_power_rcvd_LNA_500mw, ...
    r, range_power_rcvd_LNA_1w, r, range_power_rcvd_LNA_1_5w, r,
range_power_rcvd_LNA_2w);
ylabel ('Power Received (\muW)');
xlabel ('Ground Range to UAV (km)');
grid on;
Legend ('P_t = 100 mW', 'P_t = 200 mW','P_t = 500 mW', 'P_t = 1 W', 'P_t
= 1.5 W', 'P_t = 2 W');
title('Plot of Received Power vs Ground Range(With LNA)')
grid on;
```

```
figure (8);plot (r,10*log10(Gain_new))
ylabel ('Gain (dB)');
xlabel ('Ground Range to UAV (km)')
```

## D. MATLAB CODE FOR PLOTTING SCANS

```
% Plot_scans.m

%*********************************************************************
% This m file plots the output of the tracking array module and the
% pedestal angle for discrete angles. It takes in as input the filename
% and the angle computed using Sum_Beam_6elm_menu.m.
%
%written by MAJ Devieash James Pandya, Singapore, 2009.
%*********************************************************************
matrix = [];
filename = input('\n Enter in name of file to be scanned ', 's');

matrix = dlmread(filename);
angle = input('\n Enter in pedestal angle');
comput_angle = input('\n Enter in MATLAB computed angle ');

constant_angle = [];
computed_angle = [];
for i=1:length(matrix(:,1))
    constant_angle(i) = angle;
    computed_angle(i) = comput_angle;
end

axis = figure (1)
plot (matrix(:,1), matrix(:,2),matrix(:,1), constant_angle, matrix(:,1),
computed_angle)
xlabel ('Fetch Number');
ylabel('Angle (deg)');
title('Plot of Scanned Angle vs Fetch Number');
legend ('Array Scan Angle','Pedestal Angle', 'Computed AOA')
grid on;
xlim([-1 100])
ylim ([(comput_angle-6) (comput_angle+6)])
%set(figure(1)','XLim',[0 100])

figure (2)
plot (matrix(:,1), matrix(:,3))
xlabel ('Fetch Number');
ylabel('Correction Angle (deg)');
xlim([-1 100])
ylim ([(-10) (+10)])
title('Plot of Correction Angle vs Fetch Number');
legend ('Array Correction Angle')
grid on;
```

## E. MATLAB CODE FOR PLOTTING SCANS AGAINST FETCH NUMBER

```
% Plot_sweep_scans.m

%*********************************************************************
*
% This m file plots the output of the tracking array module and the
% pedestal angle against fetch number.
%
%written by MAJ Devieash James Pandya, Singapore, 2009.
%*********************************************************************
**
matrix = [];
filename = input('\n Enter in name of file to be scanned ', 's');

matrix = dlmread(filename);

constant_angle = [];


for i=1:length(matrix(:,1))
    if matrix(i,1) < 19
       constant_angle(i) = -30;
    elseif matrix(i,1) < 39
        constant_angle(i) = -25;
    elseif matrix(i,1) < 62
        constant_angle(i) = -20;
    elseif matrix(i,1) < 84
        constant_angle(i) = -15;
    elseif matrix(i,1) < 107
        constant_angle(i) = -10;
    elseif matrix(i,1) < 129
        constant_angle(i) = -5;
    elseif matrix(i,1) < 152
        constant_angle(i) = 0;
    elseif matrix(i,1) < 173
        constant_angle(i) = 5;
    elseif matrix(i,1) < 195
        constant_angle(i) = 10;
    elseif matrix(i,1) < 219
        constant_angle(i) = 15;
    elseif matrix(i,1) < 239
        constant_angle(i) = 20;
    elseif matrix(i,1) < 261
        constant_angle(i) = 25;
    else
        constant_angle(i) = 30;
    end

end

figure (1)
plot (matrix(:,1), matrix(:,2),matrix(:,1), constant_angle)
```

```
xlabel ('Fetch Number');
ylabel('Angle (deg)');
title('Plot of Scanned Angle vs Fetch Number');
legend ('Array Scan Angle','Pedestal Angle')
%xlim([-1 270])
%ylim ([-40 35])
grid on;


figure (2)
plot (matrix(:,1), matrix(:,3))
xlabel ('Fetch Number');
ylabel('Correction Angle (deg)');
title('Plot of Correction Angle vs Fetch Number');
legend ('Array Correction Angle')
xlim([-1 270])
ylim ([-40 35])
grid on;
```

## F. MATLAB CODE FOR PLOTTING SCANS FOR DIFFERING POWER LEVELS USING RSNS/DBF

```
% Plot_sweep_scans_power.m

%*********************************************************************
% This m file plots the output of the tracking array module for
% differing power levels using RSNS/DBF. The input files are formatted
% accordingly in the module.
%
%written by MAJ Devieash James Pandya, Singapore, 2009.
%*********************************************************************

matrix = [];
%filename = input('\n Enter in name of file to be scanned ', 's');

matrix_1 = dlmread('Scan_Angle_DBF_0dB.txt');
matrix_2 = dlmread('Scan_Angle_DBF_7dB.txt');
matrix_3 = dlmread('Scan_Angle_DBF_17dB.txt');
matrix_4 = dlmread('Scan_Angle_DBF_27dB.txt');
matrix_5 = dlmread('Scan_Angle_DBF_36dB.txt');
matrix_6 = dlmread('Scan_Angle_DBF_52dB.txt');
matrix_7 = dlmread('Scan_Angle_DBF_55dB.txt');

matrix = dlmread('Scan_Angle_DBF_47dB.txt');

constant_angle = [];


for i=1:length(matrix_1(:,1))
    if matrix_1(i,1) < 12
       constant_angle(i) = -30;
```

```
    elseif matrix_1(i,1) < 32
        constant_angle(i) = -25;
    elseif matrix_1(i,1) < 54
        constant_angle(i) = -20;
    elseif matrix_1(i,1) < 75
        constant_angle(i) = -15;
    elseif matrix_1(i,1) < 97
        constant_angle(i) = -10;
    elseif matrix_1(i,1) < 120
        constant_angle(i) = -5;
    elseif matrix_1(i,1) < 140
        constant_angle(i) = 0;
    elseif matrix_1(i,1) < 162
        constant_angle(i) = 5;
    elseif matrix_1(i,1) < 185
        constant_angle(i) = 10;
    elseif matrix_1(i,1) < 208
        constant_angle(i) = 15;
    elseif matrix_1(i,1) < 233
        constant_angle(i) = 20;
    elseif matrix_1(i,1) < 251
        constant_angle(i) = 25;
    else
        constant_angle(i) = 30;
    end

end

figure (1)
plot (matrix_1(:,1), constant_angle, matrix_1(:,1), matrix_1(:,2),
matrix_2(:,1), matrix_2(:,2), matrix_3(:,1), matrix_3(:,2),
matrix_4(:,1), matrix_4(:,2))
%...     matrix_5(:,1), matrix_5(:,2),matrix_6(:,1), matrix_6(:,2),
matrix_7(:,1), matrix_7(:,2))
xlabel ('Fetch Number');
ylabel('Angle (deg)');
title('Plot of Scanned Angle vs Fetch Number');
legend ('Pedestal Angle','-11 dBm','-16 dBm','-26 dBm', '-36 dBm')
xlim([-1 270])
ylim ([-40 35])
grid on;

figure (2)
plot (matrix_1(:,1), constant_angle, matrix_5(:,1),
matrix_5(:,2),matrix(:,1), matrix(:,2), matrix_6(:,1), matrix_6(:,2),
matrix_7(:,1), matrix_7(:,2))
xlabel ('Fetch Number');
ylabel('Angle (deg)');
title('Plot of Scanned Angle vs Fetch Number');
legend ('Pedestal Angle','-46 dBm','-56 dBm', '-61 dBm', '-63 dBm' )
xlim([-1 270])
ylim ([-40 35])
grid on;
```

```
figure (3)
plot (matrix_1(:,1), matrix_1(:,3))
xlabel ('Fetch Number');
ylabel('Correction Angle (deg)');
title('Plot of Correction Angle vs Fetch Number');
legend ('Array Correction Angle')
xlim([-1 270])
ylim ([-40 35])
grid on;
```

## G. MATLAB CODE FOR PLOTTING SCANS FOR DIFFERING POWER LEVELS USING RSNS

```
% Plot_sweep_scans_power.m

%*********************************************************************
% This m file plots the output of the tracking array module for
% differing power levels using RSNS. The input files are formatted
% accordingly in the module.
%
%written by MAJ Devieash James Pandya, Singapore, 2009.
%*********************************************************************
matrix = [];
%filename = input('\n Enter in name of file to be scanned ', 's');

matrix_1 = dlmread('Scan_Angle_RSNS_0.txt');
matrix_2 = dlmread('Scan_Angle_RSNS_7.txt');
matrix_3 = dlmread('Scan_Angle_RSNS_17.txt');
matrix_4 = dlmread('Scan_Angle_RSNS_27.txt');
matrix_5 = dlmread('Scan_Angle_RSNS_36.txt');
matrix_6 = dlmread('Scan_Angle_RSNS_55.txt');
matrix_7 = dlmread('Scan_Angle_RSNS_60.txt');

matrix = dlmread('Scan_Angle_RSNS_47.txt');

constant_angle = [];


for i=1:length(matrix_1(:,1))
    if matrix_1(i,1) < 18
       constant_angle(i) = -30;
    elseif matrix_1(i,1) < 36
        constant_angle(i) = -25;
    elseif matrix_1(i,1) < 55
        constant_angle(i) = -20;
    elseif matrix_1(i,1) < 72
        constant_angle(i) = -15;
    elseif matrix_1(i,1) < 90
        constant_angle(i) = -10;
```

```
    elseif matrix_1(i,1) < 109
        constant_angle(i) = -5;
    elseif matrix_1(i,1) < 128
        constant_angle(i) = 0;
    elseif matrix_1(i,1) < 146
        constant_angle(i) = 5;
    elseif matrix_1(i,1) < 164
        constant_angle(i) = 10;
    elseif matrix_1(i,1) < 182
        constant_angle(i) = 15;
    elseif matrix_1(i,1) < 201
        constant_angle(i) = 20;
    elseif matrix_1(i,1) < 218
        constant_angle(i) = 25;
    else
        constant_angle(i) = 30;
    end

end

figure (1)
plot (matrix_1(:,1), constant_angle, matrix_1(:,1), matrix_1(:,2),
matrix_2(:,1), matrix_2(:,2), matrix_3(:,1), matrix_3(:,2),
matrix_4(:,1), matrix_4(:,2))
%...     matrix_5(:,1), matrix_5(:,2),matrix_6(:,1), matrix_6(:,2),
matrix_7(:,1), matrix_7(:,2))
xlabel ('Fetch Number');
ylabel('Angle (deg)');
title('Plot of Scanned Angle vs Fetch Number');
legend ('Pedestal Angle','-11 dBm','-16 dBm','-26 dBm', '-36 dBm')
xlim([-1 270])
ylim ([-40 35])
grid on;

figure (2)
plot (matrix_1(:,1), constant_angle, matrix_5(:,1),
matrix_5(:,2),matrix(:,1), matrix(:,2), matrix_6(:,1), matrix_6(:,2),
matrix_7(:,1), matrix_7(:,2))
xlabel ('Fetch Number');
ylabel('Angle (deg)');
title('Plot of Scanned Angle vs Fetch Number');
legend ('Pedestal Angle','-46 dBm','-56 dBm', '-63 dBm', '-69 dBm' )
xlim([-1 270])
ylim ([-40 35])
grid on;

figure (3)
plot (matrix_1(:,1), matrix_1(:,3))
xlabel ('Fetch Number');
ylabel('Correction Angle (deg)');
title('Plot of Correction Angle vs Fetch Number');
legend ('Array Correction Angle')
xlim([-1 270])
```

```
ylim ([-40 35])
grid on;
```

## H. MATLAB CODE FOR PLOTTING CST SIMULATED S11 DATA AGAINST MEASURED S11 DATA

```
% compare_good_data.m

% This set of codes compares the CST subarray S11 results against the
% measured data

clear
% measured data from plot
fm=[2.3:.03:2.6];
sm=-[8.75 8.8 9 10.5 11.3 14.4 18.5 27.5 21.3 15.5 12];
% plot between frequencies f1 and f2 (in GHz)
f1=2.3;
f2=2.6;

load test0_addedS11lastrun.txt;
E=test0_addedS11lastrun;
k=0;
for i=1:length(E)
    if E(i,1)>=f1 & E(i,1)<=f2
        k=k+1; fe(k)=E(i,1); se(k)=E(i,2);
    end
end

load test0gpS11adapt255ohms.txt;
F=test0gpS11adapt255ohms;
k=0;
for i=1:length(F)
    if F(i,1)>=f1 & F(i,1)<=f2
        k=k+1; ff(k)=F(i,1); sf(k)=F(i,2);
    end
end
figure(1)
clf

% plot data for the 2 caseis in good agreement with measured
plot(ff,sf,'-k',fm,sm,'-o',fe,se,'-r')
legend('55 \Omega load','Measured','52 \Omega load')
title('Comparison of S_1_1 for the two-element subarray')
xlabel('Frequency, GHz')
ylabel('S_1_1, dB')
axis([2.3,2.6,-45,0])
```

THIS PAGE INTENTIONALLY LEFT BLANK

# LIST OF REFERENCES


[1] O. Sutton. (2003, Mission dull, dirty or dangerous? Call up a UAV: After a slow start, European industry is getting into gear to catch up Israel and the United States in today's hottest defence market. *(July - Sept),* Available: http://findarticles.com/p/articles/mi_hb3126/is_672_58/ai_n29032312/, accessed October 8, 2009.

[2] L. C. Trost, "Unmanned air vehicles (UAVs) for cooperative monitoring," Sandia National Laboratories, Tech. Rep. SAND2000–0185, 2000.

[3] R. Englehart. (2007, April 2007). UAVs are in your airspace, now. *(April 2007),* pp. 11 Nov. Available: http://www.alpa.org/portals/alpa/magazine/2007/April2007_UAVsinAirspace.pdf accessed October 12, 2009.

[4] Anonymous (2009, 4 August 2009). Microwave circuits. *2009(November / 11),* Available: http://www.ieeeghn.org/wiki/index.php/Microwave_Circuits accessed September 28, 2009.

[5] J. A. Bartee, "Genetic Algorithms as a Tool for Phased Array Radar Design," Master's Thesis, Naval Postgraduate School, Monterey, California, June 2002.

[6] G. M. Burgstaller, "Wirelessly Networked Digital Phased Array," Master's Thesis, Naval Postgraduate School, Monterey, California, September 2006.

[7] Y. Tsai, "Development of the Phase Synchronization Circuit for Wirelessly Distributed Digital Phased Array," Master's Thesis, Naval Postgraduate School, Monterey, California, September 2009.

[8] B. L. Gezer, "Multi-Beam Digital Antenna for Radar, Communications, And UAV Tracking Based On Off-The-Shelf Wireless Technologies," Master's Thesis, Naval Postgraduate School, Monterey, California, September 2006.

[9] A. Lee, "Variable Resolution Direction Finding Using the Robust Symmetrical Number System," Master's Thesis, Naval Postgraduate School, Monterey, California, December 2006.

[10] J. Benveniste, "Design and Development Of A Single Channel RSNS Direction Finder," Master's Thesis, Naval Postgraduate School, Monterey, California, March 2009.

[11] Anonymous, “Digital vs Analog,”Available: http://www.calstatela.edu/academic/comstud/jgomez/digan.htm , accessed October 4, 2009.

[12] Anonymous, “ScanEagle,” Available: http://www.boeing.com/defense-space/military/scaneagle/index.html, accessed November 30, 2009.

[13] B. King, “Why Do Different TV Standards Exist,” September 5 1995, Available: http://www.ee.surrey.ac.uk/Contrib/WorldTV/why.html, accessed October 15, 2009.

[14] National Instruments, “Anatomy of A Video Signal,” December 6 2006, Available: http://zone.ni.com/devzone/cda/tut/p/id/3020, accessed September 28, 2009.

[15] Anonymous, “NTSC Scanning, Timing/Sync, Sync Recovery, Numbers,” Available: http://www.ntsc-tv.com, accessed September 18, 2009.

[16] D. Jones, “Sync Concepts,” 2007, Available: http://www.djdesign.com/tutorial/sync.html, accessed August 10, 2009.

[17] National Instruments, “Video Signal Measurement and Generation Fundamentals,” January 2 2007, Available: http://zone.ni.com/devzone/cda/tut/p/id/4750, accessed September 17, 2009.

[18] Y. Hua, D. Iancu, J. Glossner, C. Kotlyar and A. Iancu, "Signal Processing Algorithms for DSP Implmentation of Analog TV Receivers," 2006.

[19] W. Tomasi, "*Angle Modulation Transmissions*," in *Electronic Communications Systems*, 5th Edition, pp. 254–268, Pearson Prentice Hall, Upper Saddle River, New Jersey 2004.

[20] K. M. E. Hui, "Digital Tracking Array For FM Signals Based On Off-The Shelf Wireless Technologies," 2007.

[21] D. C. Jenn, "Digital Antenna - Tracking Array," Naval Postgraduate School, Monterey California, 2006 (unpublished notes).

[22] M. Skolnik, “*Introduction to Radar Systems*” 3rd Edition, pp. 210–229, McGraw-Hill, New York, NY, 2001.

[23] D. C. Jenn, "Microwave Devices & Radar," vol. II, pp. 110–113, 2009 (lecture notes).

[24] K. Kwai, "An Analysis of Three Channel RSNS Virtual Spacing Direction Finding System," Master's Thesis, Naval Postgraduate School, Monterey, California, December 2007.

[25] D. C. Jenn, "RSNS Direction Finding Array," Naval Postgraduate School, Monterey California, 2008 (unpublished notes).

[26] D. C. Jenn, "RSNS Processing Using A Single Channel," Naval Postgraduate School, Monterey California, 2006 (unpublished notes).

[27] Anonymous, "What is I/Q Data?" April 14 2009, Available: http://zone.ni.com/devzone/cda/tut/p/id/4805 accessed September 22, 2009.

[28] C. Ziomek and P. Corredoura, "Digital I/Q Demodulator," Available: http://epaper.kek.jp/p95/ARTICLES/RPQ/RPQ02.PDF, accessed November 3, 2009.

[29] B. Razavi, "Design Considerations for Direct-Conversion Receivers," 1997, *Circuits and Systems II: Analog and Digital Signal Processing, IEEE Transactions on 44(6),* pp. 428–435.

[30] D. Brandon, D. Crook and K. Gentile, "The Advantages of Using a Quadrature Digital Upconverter (QDUC) in Point-to-Point Microwave Transmit Systems," Available: http://www.analog.com/static/imported-files/application_notes/AN-0996.pdf accessed October 20, 2009.

[31] J. Wong, M. Zou, D. Stuetzle and S. Hsiao, "Base Station Receiver Design Using a Direct-Conversion I/Q Demodulator," *High Frequency Electronics 6(2),* Available: http://www.highfrequencyelectronics.com/Archives/Feb07/HFE0207_LinearArticle.pdf, accessed October 20, 2009.

[32] A. J. Noga, "Complex Band-Pass Filters for Analytic Signal Generation and Their Application," Available: http://www.dtic.mil/cgi-bin/GetTRDoc?AD=ADA395963&Location=U2&doc=GetTRDoc.pdf accessed October 16, 2009.

[33] D. M. Pozar, "*Microwave Engineering,"* 3rd edition, Hoboken, New Jersey, J Wiley & Sons, 2005.

[34] K. Jack, "*Video Demystified: A Handbook for the Digital Engineer,"* 4th edition, Burlington, MA, Elsevier Science & Technology Books, 2004.

[35] S. A. Maas, "*Microwave Mixers,"* 2nd edition, Boston, Artech House, 1992.

[36] W. L. Stutzman and G. A. Thiele, "*Antenna Theory and Design,"* 2nd edition, New York, NY, J Wiley & Sons, 1998.

[37] Anonymous, "Application Notes for Waveguide Horns," Available: http://www.nardamicrowave.com/east/UserFiles/waveguide175.pdf, accessed October 20, 2009.

[38] B. Erenoglu, "Anechoic Chamber Evaluation," Master's Thesis, Naval Postgraduate School, Monterey, California, September 2004.

[39] J. M. Shima, "*FM Demodulation using a Digital Radio and Digital Signal Processing,"* Master's Thesis, University of Florida, Florida, 1995.

[40] K. H. Yeo, "High Resolution Spectrum Estimation for Digital Tracking Array," Master's Thesis, Naval Postgraduate School, Monterey, California, December 2009.

# INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center
   Ft. Belvoir, Virginia

2. Dudley Knox Library
   Naval Postgraduate School
   Monterey, California

3. Chairman, Code EC
   Naval Postgraduate School
   Monterey, California

4. Professor David C. Jenn
   Department of Electrical and Computer Engineering
   Naval Postgraduate School
   Monterey, California

5. Professor Philip E. Pace
   Department of Electrical and Computer Engineering
   Naval Postgraduate School
   Monterey, California

6. Robert D. Broadston
   Department of Electrical and Computer Engineering
   Naval Postgraduate School
   Monterey, California

7. Professor David Netzer
   Office of the Dean of Research
   Naval Postgraduate School
   Monterey, California

8. Kevin Jones
   Department of Mechanical & Astronautical Engineering
   Naval Postgraduate School
   Monterey, California

9. Professor Yeo Tat Soon
   Director, Temasek Defence Systems Institute (TDSI)
   National University of Singapore
   Singapore

10. Teai Yam Koon
Vice-President, ST Electronics (Satcom & Sensor Systems)
Singapore

11. Tan I-Hsiang, Leslie
ST Electronics (Satcom & Sensor Systems)
Singapore

12. MAJ Devieash James Pandya
Singapore Army
Singapore